



ITS
Institut
Teknologi
Sepuluh Nopember

PROYEK AKHIR - VE 180626

**PENGUNAAN SENSOR *FINGERPRINT* SEBAGAI
AKTIVASI MESIN KONVEYOR DAN PENDATAAN
BARANG SECARA OTOMATIS**

Pratiwi Maisita Sari
NRP 10311600000027

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng.
Enny Indasyah, S.ST., M.T., M.Sc.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019



FINAL PROJECT - VE 180626

***FINGERPRINT SENSOR AS CONVEYOR MACHINE
ACTIVATION AND AUTOMATIC DATA COLLECTION***

Pratiwi Maisita Sari
NOR 10311600000027

Supervisor
Ir. Josaphat Pramudijanto, M.Eng.
Enny Indasyah, S.ST., M.T., M.Sc.

***DEPARTMENT OF ELECTRICAL ENGINEERING AUTOMATION
Vocational Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2019***

PERNYATAAN KEASLIAN PROYEK AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Proyek Akhir saya dengan judul "**Penggunaan Sensor *Fingerprint* sebagai Aktivasi Mesin Konveyor dan Pendataan Barang Secara Otomatis**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 03 Juli 2019



Pratiwi Maisita Sari
NRP 10311600000027

-----Halaman ini sengaja dikosongkan-----

**PENGUNAAN SENSOR *FINGERPRINT* SEBAGAI
AKTIVASI MESIN KONVEYOR DAN PENDATAAN
BARANG SECARA OTOMATIS**

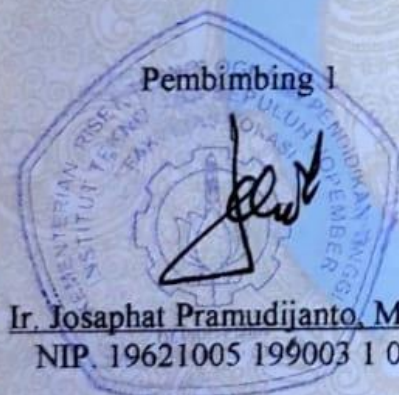
PROYEK AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Ahli Madya Teknik
Pada
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui :

Pembimbing 1

Pembimbing 2



Ir. Josaphat Pramudijanto, M.Eng.
NIP. 19621005 199003 1 003

Enny Indasyah, S.ST., M.T., M.Sc.
NIP. 199103302018072001

**SURABAYA
JULI, 2019**

-----Halaman ini sengaja dikosongkan-----

PENGUNAAN SENSOR *FINGERPRINT* SEBAGAI AKTIVASI MESIN KONVEYOR DAN PENDATAAN BARANG SECARA OTOMATIS

Nama Mahasiswa : Pratiwi Maisita Sari
NRP : 10311600000027
Dosen Pembimbing 1 : Ir. Josaphat Pramudijanto, M.Eng
NIP : 19621005 199003 1 003
Dosen Pembimbing 2 : Enny Indasyah, S.S.,MT.,M.Sc.
NIP : 199103302018072001

ABSTRAK

Saat ini banyak industri yang sudah menggunakan conveyor untuk melakukan proses distribusi. Kegiatan yang dilakukan umumnya dengan cara manual. Oleh karena itu, dibutuhkan suatu alat yang dapat mempercepat dan meringankan pekerjaan manusia, serta dapat meminimalisir terjadinya kesalahan yang terjadi pada industri manual.

Untuk itu, dibuatlah alat yang dapat mengaktifkan mesin konveyor menggunakan *fingerprint* dan mengelompokkan serta menjumlahkan barang secara otomatis. Cara kerja alat ini yaitu karyawan harus *login* dengan *fingerprint* terlebih dahulu. Setelah *login* berhasil, maka mesin konveyor akan menyala. Setelah itu barang yang melewati konveyor dan terdeteksi oleh sensor ketinggian, maka akan dikelompokkan dan terhitung secara otomatis sesuai dengan jenis barang. Ada 3 jenis barang yaitu kecil, sedang dan besar. Kemudian untuk mematikan mesin konveyor ini harus *logout* dengan menempelkan jari yang sama lagi ke sensor *fingerprint*. Data yang didapat akan diproses oleh arduino mega dan hasil datanya akan ditampilkan pada komputer menggunakan *software* PLX DAQ dan jumlah barang juga ditampilkan pada LCD.

Setelah dilakukan pengujian, hasil dari pengujian menunjukan bahwa keberhasilan pembacaan sensor *fingerprint* untuk aktivasi mesin yaitu sebesar 92,5 %. Kesalahan pembacaan sensor *fingerprint* ini disebabkan karena jari yang kotor, pada pengujian ini saya menggunakan tangan yang sedikit berminyak. Sedangkan keberhasilan dari pengujian sensor dioda laser yaitu sebesar 100%. Karena berhasil mendeteksi barang dan mengelompokkannya dengan tepat.

Kata Kunci : Konveyor, *Fingerprint*, Arduino Mega

-----Halaman ini sengaja dikosongkan-----

**FINGERPRINT SENSOR AS CONVEYOR MACHINE
ACTIVATION AND AUTOMATIC DATA COLLECTION**

Student Name : Pratiwi Maisita Sari
Registration Number : 10311600000027
Supervisor 1 : Ir. Josaphat Pramudijanto, M.Eng
ID : 19621005 199003 1 003
Supervisor 2 : Enny Indasyah, S.S.,MT.,M.Sc.
ID : 199103302018072001

ABSTRACT

Today many industries have used conveyors to carry out the distribution process. Activities carried out generally by manual method. Therefore, we need a tool that can speed up and alleviate human work, and can minimize the occurrence of errors that occur in the manual industry.

For this reason, a tool is made that can activate the conveyor machine using fingerprint and group and sum up the items automatically. The way this tool works is that the employee must log in with the fingerprint first. After the login is successful, the conveyor engine will turn on. After that the item passes through the conveyor and is detected by a height sensor, it will be grouped and counted automatically according to the type of item. There are 3 types of goods, namely small, medium and large. Then to turn off the conveyor engine it must logout by attaching the same finger again to the fingerprint sensor. The data obtained will be processed by Arduino Mega and the data results will be displayed on the computer using the PLX DAQ software and the number of items also displayed on the LCD

After testing, the results of the tests show that the success of fingerprint sensor readings for machine activation is 92.5%. This fingerprint sensor reading error is caused by dirty fingers, in this test I used slightly oily hands. While the success of testing laser diode sensors is 100%. Because it successfully detects items and classifies them correctly.

Keywords : Conveyors, Fingerprint, Arduino Mega

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa yang telah memberikan rahmat dan karunianya, sehingga kami dapat menyelesaikan Proyek Akhir ini dengan baik, begitu pula dengan pembuatan buku Proyek Akhir ini.

Proyek Akhir ini dilakukan untuk memenuhi beban satuan kredit semester (SKS) yang harus ditempuh sebagai persyaratan akademis di Departemen Teknik Elektro Otomasi Fakultas Vokasi Institut Teknologi Sepuluh Nopember Surabaya untuk menyelesaikan program pendidikan Diploma di Teknik Elektro dengan judul :

“Penggunaan Sensor *Fingerprint* sebagai Aktivasi Mesin Konveyor dan Pendataan Barang Secara Otomatis ”

Dengan terselesainya Proyek Akhir ini, kami mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Kedua orang tua yang senantiasa mendoakan dan memberikan dukungan dengan tulus tiada henti.
2. Bapak Ir. Joko Susila, M.T. selaku Kepala Departemen Teknik Elektro Otomasi
3. Bapak Ir. Josaphat Pramudijanto, M.Eng. selaku dosen pembimbing 1, yang telah berkenan memberikan pengarahan dan bimbingan dari awal hingga terselesainya Proyek Akhir.
4. Ibu Enny Indasyah, S.S.,MT.,M.Sc. selaku dosen pembimbing 2, yang tidak pernah bosan memberikan bimbingan.
5. Teman-teman D3 Teknik Elektro Foxfire yang telah membantu dan mendukung dalam penyelesaian Proyek Akhir.
6. Semua pihak yang telah membantu penyelesaian Proyek Akhir, baik secara langsung maupun tidak langsung .

Penulis menyadari dan memohon maaf atas segala kekurangan pada Proyek Akhir ini. Akhir kata, semoga Proyek Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 03 Juli 2019

Pratiwi Maisita Sari

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL	i
PERNYATAAN KEASLIAN PROYEK AKHIR	v
LEMBAR PENGESAHAN	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xxi
 BAB I PENDAHULUAN	 1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Sistematika Laporan	3
1.6 Relevansi	4
 BAB II TEORI DASAR	 5
2.1 Konveyor	5
2.1.1 Jenis – Jenis Konveyor	5
2.2 Sensor <i>Fingerprint</i>	7
2.3 Sensor Dioda Laser	10
2.4 Arduino Mega 2560	11
2.5 <i>Relay</i>	12
2.5.1 Jenis-jenis <i>Relay</i>	13
2.5.2 Modul <i>Relay</i> Arduino	14
2.6 Motor DC	15
2.7 LCD 20 x 4	16
2.8 <i>Software</i> Arduino IDE	16
2.9 <i>Software</i> PLX DAQ	17
 BAB III PERANCANGAN <i>HARDWARE</i> DAN <i>SOFTWARE</i>	 19
3.1 Perancangan <i>Hardware</i>	20
3.1.1 Perancangan Konveyor	20
3.1.2 Perancangan Contoh Barang	22

3.1.3	<i>Wiring Fingerprint</i>	22
3.1.4	<i>Wiring Photo Dioda</i>	23
3.1.5	<i>Wiring LCD 20x4</i>	24
3.1.6	<i>Wiring Modul Relay</i>	25
3.1.7	<i>Wiring Push Button ke Arduino</i>	29
3.1.6	Hasil <i>Wiring</i> Secara Keseluruhan.....	30
3.1.7	Perancangan <i>Panel Box</i>	31
3.1.8	Perancangan Regulator Tegangan	32
3.2	Perancangan <i>Software</i>	32
3.2.1	<i>Flowchart Login</i> untuk Aktivasi Mesin Konveyor	34
3.2.2	<i>Flowchart</i> Pengelompokan Barang Berdasarkan Ketinggian.....	35
3.2.3	<i>Flowchart Logout</i> untuk Mematikan Konveyor	40
3.2.4	<i>Flowchart</i> Penambahan Sidik Jari Baru.....	41
3.2.5	Perancangan Pendataan Barang Menggunakan <i>Software</i> PLX DAQ.....	43
BAB IV PENGUJIAN DAN ANALISA		47
4.1	Pengujian <i>Fingerprint Scanner</i>	47
4.2	Pengujian LCD 20x4.....	49
4.3	Pengujian Penambahan Sidik Jari Baru.....	53
4.4	Pengujian Waktu Sensor Dioda Laser Mendeteksi Barang	67
4.5	Pengujian Sistem Keseluruhan.....	69
BAB V PENUTUP.....		86
5.1	Kesimpulan.....	87
5.2	Saran	88
DAFTAR PUSTAKA.....		89
LAMPIRAN A.....		A-1
LAMPIRAN B.....		B-1
LAMPIRAN C.....		C-1
DAFTAR RIWAYAT HIDUP.....		D-1

DAFTAR GAMBAR

HALAMAN

Gambar 2.1	<i>Belt Konveyor</i>	6
Gambar 2.2	<i>Roller Konveyor</i>	6
Gambar 2.3	<i>Chain Konveyor</i>	7
Gambar 2.4	<i>Hanging Over Head Konveyor</i>	7
Gambar 2.5	Jenis Umum Pola Metode Minutia	8
Gambar 2.6	Sensor <i>Fingerprint</i> dan Konektor	9
Gambar 2.7	Bentuk Fisik Modul Sensor Dioda Laser.....	11
Gambar 2.8	Arduino Mega.....	12
Gambar 2.9	Bentuk Fisik dan Simbol <i>Relay</i>	12
Gambar 2.10	Jenis <i>Relay</i> Berdasarkan <i>Pole</i> dan <i>Throw</i>	13
Gambar 2.11	Modul <i>Relay</i> Arduino 8 <i>Channel</i>	14
Gambar 2.12	Modul <i>Relay</i> Arduino 2 <i>Channel</i>	14
Gambar 2.13	Motor DC	15
Gambar 2.14	LCD 20x4.....	16
Gambar 2.15	<i>Sketch</i> dari <i>Software</i> Arduino IDE	17
Gambar 2.16	Tampilan pada PLX DAQ	18
Gambar 3.1	Diagram Fungsional Cara Kerja Konveyor	19
Gambar 3.2	Rancangan Konveyor	21
Gambar 3.3	Penampang dari Contoh Barang.....	22
Gambar 3.4	<i>Wiring Fritzing</i> <i>Fingerprint</i> ke Arduino Mega	22
Gambar 3.5	<i>Wiring Fritzing</i> <i>Photo Dioda</i> ke Arduino Mega.....	23
Gambar 3.6	<i>Wiring Fritzing</i> LCD 20x4 ke Arduino Mega.....	25
Gambar 3.7	<i>Wiring Fritzing</i> Arduino Mega ke <i>Relay</i>	25
Gambar 3.8	<i>Wiring Fritzing</i> Modul <i>Relay</i> ke Motor	26
Gambar 3.9	<i>Wiring Fritzing</i> <i>Power Supply</i> ke Modul <i>Relay</i>	29
Gambar 3.10	<i>Wiring Fritzing</i> <i>Push Button</i> ke Arduino.....	30
Gambar 3.11	<i>Wiring</i> Secara Keseluruhan	30
Gambar 3.12	<i>Design</i> dari <i>Panel Box Fingerprint</i> , <i>LCD</i> , dan <i>Button</i>	31
Gambar 3.13	<i>Design</i> dari Panel Sensor Dioda Laser	31
Gambar 3.14	Rangkaian Regulator Tegangan	32
Gambar 3.15	<i>Flowchart</i> Secara Keseluruhan	33
Gambar 3.16	<i>Flowchart</i> Login untuk Mengaktifkan Konveyor.....	34
Gambar 3.17	<i>Flowchart</i> Pengelompokan Barang.....	35
Gambar 3.18	<i>Flowchart</i> STATE A.....	37
Gambar 3.19	<i>Flowchart</i> STATE B.....	38

Gambar 3.20 <i>Flowchart</i> STATE C	39
Gambar 3.21 <i>Flowchart Logout</i> untuk Mematikan Konveyor	41
Gambar 3.22 <i>Flowchart</i> Penambahan Sidik jari Baru	42
Gambar 3.23 <i>Connect</i> PLX DAQ ke Arduino	44
Gambar 3.24 Perancangan Data pada <i>Software</i> PLX DAQ	44
Gambar 4.1 Program <i>Enroll</i> untuk Daftar Sidik Jari	47
Gambar 4.2 Pengujian <i>Enrollment</i> pada <i>Fingerprint Scanner</i>	48
Gambar 4.3 Program untuk Mengecek Kecocokan Sidik Jari.....	48
Gambar 4.4 Pengujian Kecocokan Sidik Jari	49
Gambar 4.5 Program LCD untuk Tampil Menu Sebelum <i>Login</i>	50
Gambar 4.6 Tampilan Awal LCD untuk <i>Login</i>	50
Gambar 4.7 Tampilan LCD Ketika <i>Login</i> Berhasil	51
Gambar 4.8 Tampilan LCD Ketika <i>Login</i> Gagal.....	51
Gambar 4.9 Program Tampilan LCD Jumlah Barang.....	52
Gambar 4.10 Tampilan LCD saat Mendeteksi Barang Kecil	52
Gambar 4.11 Tampilan Ketika Semua Sensor Mendeteksi.....	53
Gambar 4.12 Program untuk Menambahkan Sidik Jari Baru.....	54
Gambar 4.13 Ilustrasi Pemilihan Menu “ <i>Add User</i> ”	54
Gambar 4.14 Ilustrasi Pemilihan NIP	55
Gambar 4.15 Pemilihan menu “ <i>Next</i> ” dengan Tombol C	55
Gambar 4.16 Proses Admin <i>Scanning</i> Sidik Jari.....	56
Gambar 4.17 <i>Scanning</i> Sidik Jari Baru Pertama	56
Gambar 4.18 <i>Scanning</i> Sidik Jari Baru Kedua.....	56
Gambar 4.19 Pemberitahuan Sidik Jari Berhasil Terdaftar	57
Gambar 4.20 Posisi Sidik Jari saat Lurus	58
Gambar 4.21 Posisi Sidik Jari Saat Miring Kiri	58
Gambar 4.22 Posisi Sidik Jari saat Miring Kanan	58
Gambar 4.23 Program Deteksi Barang Berdasarkan Ketinggian	65
Gambar 4.24 Salah Satu Pengujian Jarak Deteksi Sensor	65
Gambar 4.25 Tampilan LCD Saat Sensor Mendeteksi Barang	67
Gambar 4.26 Salah Satu Pengujian Waktu Deteksi Sensor	68
Gambar 4.27 Tampilan Awal Layar LCD 20x4.....	70
Gambar 4.28 Tampilan Awal Perintah <i>Login</i> di LCD	70
Gambar 4.29 Tampilan LCD Ketika <i>Login</i> Berhasil.....	71
Gambar 4.30 Program Pengelompokkan Barang Kecil	71
Gambar 4.31 Barang Kecil saat Terdeteksi Oleh Sensor 1	72
Gambar 4.32 Barang Kecil Terdeteksi Sensor Ketinggian	72
Gambar 4.33 Barang Kecil Melewati Konveyor 2 dan Naik Ke <i>Lift</i>	73
Gambar 4.34 Barang Melewati Konveyor 4	74

Gambar 4.35 Tampilan LCD saat Mendeteksi Barang Kecil	74
Gambar 4.36 Program Mengelompokkan Barang Sedang.....	75
Gambar 4.37 Barang Sedang Terdeteksi oleh Sensor di Konveyor 1....	76
Gambar 4.38 Barang Sedang Terdeteksi oleh Sensor Ketinggian.....	76
Gambar 4.39 Barang Sedang Akan Didorong oleh <i>Pusher</i>	77
Gambar 4.40 Barang Sedang Jatuh didorong oleh <i>Pusher</i>	77
Gambar 4.41 Tampilan saat Barang Sedang Terdeteksi Sensor	78
Gambar 4.42 Program Mengelompokkan Barang Besar	79
Gambar 4.43 Barang Besar Terdeteksi Sensor di Konveyor 1	79
Gambar 4.44 Barang Besar Terdeteksi Sensor Dioda Laser.....	80
Gambar 4.45 Barang Besar di Konveyor 3.....	80
Gambar 4.46 Barang Besar Terjatuh Kebawah	81
Gambar 4.47 Tampilan LCD saat Barang Terdeteksi Sensor	81
Gambar 4.48 Program untuk <i>Logout</i>	82
Gambar 4.49 Tampilan Perintah <i>Logout</i> pada LCD.....	83
Gambar 4.50 Tampilan Ketika <i>Logout</i> Berhasil	83
Gambar 4.51 Program Menampilkan Data pada PLX DAQ.....	84
Gambar 4.52 Data Tersimpan pada PLX DAQ	84

-----Halaman ini sengaja dikosongkan-----

DAFTAR TABEL

	HALAMAN
Tabel 2.1 Spesifikasi <i>Fingerprint</i>	9
Tabel 2.2 Spesifikasi Arduino Mega.....	11
Tabel 2.3 Spesifikasi Modul <i>Relay</i> Arduino.....	14
Tabel 2.4 Spesifikasi Motor DC	15
Tabel 2.5 Spesifikasi Kaki pada LCD 16x4.....	16
Tabel 3.1 Ukuran dari Contoh Barang	22
Tabel 3.2 Keterangan <i>Wiring</i> Fritzing Fingerprint ke Arduino mega...	23
Tabel 3.3 Keterangan <i>Wiring</i> Fritzing <i>Photo</i> Dioda ke Arduino mega	23
Tabel 3.4 Keterangan <i>Wiring</i> Fritzing dari LCD 20x4 ke Arduino.....	25
Tabel 3.5 Keterangan <i>Wiring</i> Fritzing dari Arduino Mega ke <i>Relay</i> ...	26
Tabel 3.6 Keterangan <i>Wiring</i> Fritzing Modul <i>Relay</i> ke Motor	27
Tabel 3.7 Keterangan <i>Wiring</i> Fritzing <i>Relay</i> ke Motor untuk <i>Lift</i>	27
Tabel 3.8 Keterangan <i>Wiring</i> Fritzing <i>Relay</i> ke Motor untuk <i>Pusher</i> .	28
Tabel 3.9 Keterangan <i>Wiring</i> Fritzing <i>Relay</i> ke <i>Power supply</i>	29
Tabel 4.1 Hasil Pengujian Berdasarkan Posisi Sidik Jari	59
Tabel 4.2 Hasil Pengujian Berdasarkan Kondisi Sidik Jari	62
Tabel 4.3 Hasil Pengujian Ketepatan Deteksi Sensor Rendah.....	66
Tabel 4.4 Hasil Pengujian Ketepatan Deteksi Sensor Sedang	66
Tabel 4.5 Hasil Pengujian Ketepatan Deteksi Sensor Tinggi	66
Tabel 4.6 Hasil Pengujian Waktu Deteksi Sensor.....	68
Tabel 4.7 Pengujian Kecepatan Naik <i>Lift</i> Berdasarkan Berat Barang .	73
Tabel 4.8 Hasil Pengujian Pengelompokkan Barang Kecil	75
Tabel 4.9 Hasil Pengujian Pengelompokkan Barang Sedang	78
Tabel 4.10 Hasil Pengujian Pengelompokkan Barang Besar.....	81

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan jaman yang semakin pesat, kebutuhan akan efektifitas dan efisiensi sangat diutamakan dalam berbagai bidang. Pada suatu perusahaan awalnya masih dilakukan secara manual, seperti contoh di perusahaan industri minuman. Pemindahan dari botol yang sudah diisi minuman untuk dikemas masih dilakukan secara manual. Yaitu masih dipindahkan oleh manusia untuk dilanjutkan ke proses pengemasan. Hal ini masih banyak kekurangan pada industri manual, salah satunya membutuhkan waktu pengerjaan yang lama. Sehingga dalam proses pendistribusian dibutuhkan alat yang dapat memindahkan produk dari tempat satu ke tempat lainnya dengan cepat yaitu dengan konveyor. Dengan konveyor, manusia cukup meletakkan barang di konveyor sehingga barang otomatis berjalan sampai tempat yang dituju. Hal ini lebih memudahkan dan mempercepat proses pengerjaan.

Pada umumnya, di perusahaan yang sudah menerapkan penggunaan konveyor proses aktivasi mesinnya pun masih dilakukan secara manual. Pengaktifan mesin konveyor dilakukan secara manual dengan menggunakan tombol *on/off* yang dilakukan oleh seorang operator mesin. Sehingga setelah pengerjaan selesai, operator mesin harus melaporkan bagaimana kondisi alatnya, berapa lama mereka bekerja jika ada lembur, jenis produk, banyak produk yang dihasilkan, dll. Dengan ini, semua orang bisa menyalakan mesin konveyor tersebut walaupun bukan operator mesin tersebut. Jadi, tidak bisa diketahui siapa orang yang sedang mengoperasikan mesin tersebut. Sehingga dibutuhkan alat yang dapat meminimalisir terjadinya kecurangan dan alat ini dapat mempermudah untuk melakukan pendataan waktu seorang operator mesin selama ia bekerja. Yaitu mengaktifkan mesin konveyor menggunakan *fingerprint*, supaya tidak semua orang bisa mengaktifkan konveyor tersebut. Jadi, ketika *login* pada *fingerprint* berhasil, maka otomatis mesin akan menyala dan otomatis data tersebut akan dikirimkan ke komputer. Untuk mematikannya pun harus *logout* terlebih dahulu. Jadi dapat memudahkan pendataan kehadiran maupun keterlambatan karyawan bahkan karyawan yang bekerja lebih atau lembur. Sehingga memudahkan perusahaan untuk mendapatkan data karyawan yang telah mengoperasikan mesin konveyor.

Permasalahan selanjutnya yang dihadapi oleh suatu perusahaan yaitu saat pengemasan produk. Mengelompokkan produk sesuai dengan jenisnya biasa dilakukan dengan cara manual. Contohnya di suatu perusahaan air minum yaitu aqua, disitu botol aqua berbeda ukuran ada botol kecil, sedang dan besar. Untuk melakukan pengemasan kita harus memilah satu persatu botol aqua untuk dikelompokkan sesuai dengan jenisnya. Untuk menghitung total barang berdasarkan jenisnyapun harus menghitung satu persatu dan dicatat pada kertas. Namun apabila para pekerja pabrik harus melakukan semua itu secara manual, hal itu akan memakan waktu yang cukup lama dan tidak praktis. Hal itu juga dapat menyebabkan terjadinya kesalahan pendataan karena kurang teliti. Maka dibutuhkan suatu alat yang dapat mengelompokkan barang sesuai dengan jenisnya dan dapat menghitung jumlah barang secara otomatis. Yaitu kami menambahkan suatu fitur pada konveyor dengan sensor ketinggian menggunakan dioda laser. Jadi, barang yang telah melewati konveyor dan terdeteksi oleh sensor ketinggian maka otomatis dapat mengetahui jenis barang yang kemudian barang dikelompokkan berdasarkan jenisnya. Alat ini juga dapat menghitung secara otomatis yang ditampilkan pada LCD serta data tersimpan pada komputer. Jadi lebih efisien untuk melakukan proses pendataan.

Prototype konveyor ini dikembangkan dari alat yang sudah ada sebelumnya yaitu milik Andrik Kurnia Pratama dan Rahmat Bagus Prasojodi tahun 2015, dengan judul Proyek Akhir “Mesin Pengelompok Barang Berdasarkan Ketinggian dan Konveyor Pengangkut Menggunakan PLC”. Begitu pula yang dikembangkan lagi dari judul sebelumnya yaitu milik Nuva Choironi Ersha tahun 2018, dengan judul Proyek Akhir Akhir “Monitoring Program Pengelompokkan dan Penghitung Jumlah Barang pada Konveyor Menggunakan Mikrokontroler”.

1.2 Permasalahan

Dalam sistem distribusi pada industri yang menggunakan konveyor, masih sering terjadi kesalahan pendataan baik dalam hal pendataan karyawan yang telah mengoperasikan mesin konveyor maupun kesalahan mengelompokkan serta menghitung jumlah barang yang telah diselesaikan. Hal ini dirasa masih kurang efektif karena harus mengelompokkan barang satu persatu dan menghitungnya secara manual.

1.3 Batasan Masalah

Batasan Masalah dari Proyek Akhir ini yaitu :

1. Barang yang akan dikelompokkan berdasarkan ketinggian barang yaitu hanya 3 jenis barang. Barang kecil, sedang dan besar.
2. Untuk berat barang yang menaiki *lift* idealnya yaitu dengan berat 14 gram (*box* barang dari kertas *duplex*).

1.4 Tujuan

Tujuan yang akan dicapai dari Proyek Akhir ini adalah:

1. Membuat suatu sistem otomatis yang dapat mengaktifkan mesin konveyor menggunakan *fingerprint*.
2. Membuat sistem otomatis yang dapat menghitung serta mengelompokkan barang sesuai dengan jenisnya.
3. Mampu menyajikan data karyawan serta jumlah barang yang terdeteksi melalui layar LCD dan menyimpannya pada *software* PLX DAQ.

1.5 Sistematika Laporan

Pembahasan Proyek Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, batasan masalah, tujuan, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Bab ini membahas teori dasar, teori penunjang dari peralatan-peralatan yang digunakan dalam pembuatan alat.

Bab III Perancangan Sistem

Bab ini membahas mengenai desain perancangan dan pembuatan perangkat keras (*hardware*) yang terdiri dari atas perancangan rangkaian elektronika, perancangan mekanik serta perangkat lunak (*software*) yang terdiri atas program untuk mengaktifkan mesin konveyor menggunakan sensor *fingerprint* dan menghitung jumlah barang yang lewat pada konveyor menggunakan sensor dioda laser.

Bab IV Pengujian dan Analisa Sistem

Bab ini membahas tentang pengukuran, pengujian alat dan analisa data yang didapat dalam pengujian alat.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.6 Relevansi

Manfaat atau relevansi dari Proyek Akhir ini dapat diuraikan yaitu alat ini dapat digunakan untuk alat bantu atau media pembelajaran mengenai teknologi di industri. Baik dari sistem penggunaan *fingerprint* sebagai aktivasi mesin ataupun pengelompokkan barang serta perhitungan jumlah barang secara otomatis. Dari alat yang akan dibuat ini diharapkan berhasil untuk mengaktifkan mesin konveyor dari sidik jari yang berbeda yang telah terdaftar pada sensor *fingerprint*. Selain itu alat ini juga dapat mengelompokkan jenis barang berdasarkan ketinggian yang dideteksi oleh sensor dioda laser dengan tepat dan menghitung secara otomatis. Ketika permodelan sistem ini berhasil, baik dari pengaktifan mesin konveyor menggunakan *fingerprint* dan pengelompokkan barang yang dihasilkan sudah tepat sesuai dengan yang diinginkan, dalam jangka panjang dapat dikembangkan untuk skala besar di industri sebenarnya.

BAB II

TEORI DASAR

Bab ini membahas mengenai teori penunjang dari peralatan yang digunakan dalam Penggunaan Sensor *Fingerprint* sebagai Aktivasi Mesin dan Pendataan Barang Secara Otomatis pada Konveyor Menggunakan Mikrokontroler seperti Sensor *Fingerprint*, Arduino Mega, *Relay*, Motor DC, Sensor Dioda Laser, PLX DAQ dan lain sebagainya

2.1 Konveyor [1]

Konveyor adalah suatu sistem mekanik yang mempunyai fungsi memindahkan barang dari satu tempat ke tempat yang lain. Konveyor banyak dipakai di industri untuk transportasi barang yang jumlahnya sangat banyak dan berkelanjutan. Konveyor terutama berguna dalam aplikasi yang melibatkan transportasi bahan berat ataubesar. Sistem konveyor memungkinkan transportasi cepat dan efisien untuk berbagai bahan. Banyak jenis sistem konveyor yang tersedia, dan digunakan sesuai dengan kebutuhan berbagai industri yang berbeda. Dalam kondisi tertentu, konveyor banyak dipakai karena mempunyai nilai ekonomis dibanding transportasi berat seperti truk dan mobil pengangkut. Konveyor dapat memobilisasi barang dalam jumlah banyak dan kontinyu dari satu tempat ke tempat lain

2.1.1 Jenis – Jenis Konveyor

Ada banyak macam jenis konveyor yang ada saat ini diantaranya adalah :

- *Belt* Konveyor

Konveyor jenis ini dilengkapi dengan sabuk sebagai alat angkutnya. Sabuk yang digunakan juga disesuaikan dengan material yang diangkut. Konveyor ini menggunakan motor sebagai penggerak utama yang terhubung dengan drum yang disebut *pulley*, *pulley* inilah yang yang diselubungi oleh *belt* yang lebarnya sama dengan *pulley* tersebut dan panjangnya *belt* menyesuaikan dengan kebutuhan atau kapasitas angkut serta jarak angkut material tersebut. Jika motor dijalankan maka *pulley* akan ikut berputar seiring motor hingga *belt* yang menyelubungi ikut bergerak tertarik kearah putaran

drum atau *pulley* tersebut. Untuk dapat mengetahui bentuk dari *belt* konveyor ini dapat dilihat pada Gambar 2.1.



Gambar 2.1 *Belt Konveyor* [1]

- *Roller Konveyor*
Roller Konveyor ini adalah konveyor yang paling umum digunakan karena lintasan geraknya tersusun dari beberapa tabung (*roll*) yang tegak lurus terhadap arah lintasannya di mana plat datar yang ditempatkan untuk menahan beban akan bergerak sesuai dengan arah putaran *roll*. Untuk melihat bentuk dari *roller* konveyor ini dapat dilihat pada Gambar 2.2.



Gambar 2.2 *Roller Konveyor* [1]

- *Chain Konveyor*
Konveyor jenis ini menggunakan rantai sebagai penggerak utama material. Rangkaian rantai yang tidak terputus satu dengan yang lain juga berbentuk sekat-sekat, maka konveyor ini dapat membawa material yang akan dipindahkan. Konveyor jenis ini biasanya digunakan untuk

mengangkut material yang tidak berdimensi, seperti bahan tambang ataupun kulit kelapa sawit yang akan diolah oleh industri. Untuk bentuk fisik dari *Chain Konveyor* ini dapat dilihat pada Gambar 2.3.



Gambar 2.3 *Chain Konveyor* [1]

- *Hanging Over Head Konveyor*
Konveyor ini berbeda dengan yang lain dikarenakan dalam memindahkan barangnya dengan cara digantung. Tenaga penggerak dari konveyor ini menggunakan gaya gravitasi ataupun motor. Untuk bentuk fisik dari *hanging over head* konveyor ini bisa dilihat pada Gambar 2.4.

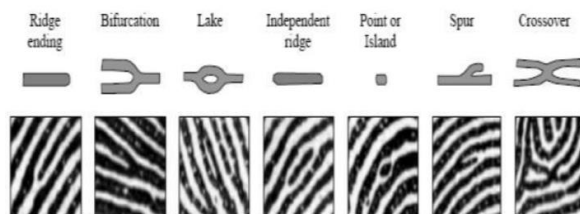


Gambar 2.4 *Hanging Over Head Konveyor* [1]

2.2 **Sensor *Fingerprint*** [2]

Sensor *Fingerprint* atau alat pemindai sidik jari merupakan suatu komponen elektronik yang berfungsi membaca pola sidik jari dan mencocokkan dengan *database* yang telah tersimpan dalam memori. Sistem ini memiliki cara kerja yang terbagi menjadi tiga yaitu *enrollment*, *delete*, dan *fingerprint access*. Pada saat ini, sensor sidik jari telah banyak dipasarkan. Divisi forensik pada penegak hukum saat

ini menggunakan sidik jari sebagai proses penyelidikan. Hal ini dikarenakan setiap orang memiliki garis atau bentuk sidik jari yang berbeda – beda. Pola pembacaan sidik jari dapat dibandingkan dengan metode minutiea. Metode ini sering digunakan sebagai metode perbandingan sensor *fingerprint* berdasarkan pencocokan pola minutiea. Minutiea yang berarti kecil, dalam konteks sidik jari hal tersebut berarti pola – pola pada sidik jari. Sebagian besar metode ini memerlukan gambar sidik jari untuk dikonversi menjadi gambar biner.



Gambar 2.5 Jenis Umum Pola Metode Minutiea [2]

Pada saat sistem memulai *scanning* sidik jari dengan alat pemindai, hasil akan disimpan dalam memori. Hasil *scanning* disimpan dalam format digital pada saat *enrollment* (pendaftaran sidik jari). Setelah itu, rekaman sidik jari tersebut diproses dan dibuatkan daftar pola fitur sidik jari yang unik. Pola fitur sidik jari yang unik tersebut kemudian disimpan dalam memori atau *database*. Pola sidik jari yang unik ini disebut dengan pola minutiea. Pada saat identifikasi, pola minutiea tersebut kemudian dicocokkan dengan hasil *scan* sidik jari. Teknik pembacaan sidik jari oleh mesin absensi sidik jari dapat dikelompokkan menjadi empat tipe, yaitu teknik optik, ultrasonik, kapasitansi dan thermal. Kelemahan metode ini adalah hasil *scanning* sangat tergantung dari kualitas sidik jari. Jika kualitas sidik jari luka, kering, dan tidak jelas, maka kualitas hasil pembacaan tidak bagus. Kelemahan lain adalah teknik ini bisa diakali dengan jari palsu. Tapi teknik ini mempunyai keuntungan mudah dilakukan dan tidak membutuhkan biaya yang mahal.



Gambar 2.6 Sensor *Fingerprint* dan Konektor [2]

Pada Gambar 2.6 di atas merupakan contoh modul pemindai sidik jari yang dapat dikoneksikan dengan Arduino. Dengan DSP prosesor yang berkecepatan tinggi, modul *fingerprint scanner* ini juga dapat diaplikasikan dengan serial *device* yang lain, seperti MSP430, AVR, PIC, STM32, ARM dan FPGA *device*. Modul ini memiliki memori yang dapat menyimpan hingga 300 data sidik jari. Sensor *fingerprint* memiliki kemampuan pembacaan sidik jari dengan tingkat sensitivitas yang tinggi baik dalam keadaan basah. Selain itu alat ini memiliki kecepatan tinggi saat melakukan sistem pemindaian, pencarian dan perbandingan pola sidik jari.

Dengan fitur-fitur dan segala keunggulan tersebut, *fingerprint scanner* ini dapat difungsikan dalam berbagai bidang, terutama yang bersangkutan dengan masalah keamanan. Dapat difungsikan sebagai piranti absensi modern, saklar elektronik maupun pengganti password dengan sensitivitas yang tinggi. Untuk melihat spesifikasi *fingerprint* dapat dilihat pada Tabel 2.1

Tabel 2.1 Spesifikasi *Fingerprint*

Spesifikasi	Keterangan
<i>Supply Voltage</i>	3,6-6 VDC/ 3.3 V <i>Supplying</i>
<i>Operating Current</i>	<120 Ma
<i>Peak Current</i>	<140 mA
<i>Fingerprint imaging time</i>	<1.0 seconds
<i>Window area</i>	14 mm x 18 mm
<i>Signature file</i>	256 bytes
<i>Template file</i>	512 bytes
<i>Storage capacity</i>	300 templates
<i>False Reject rate</i>	<0.001 % (<i>Security level 3</i>)
<i>PC Interface</i>	UART (TTL <i>logic level</i>) or USB 2.0/USB 1.1

Spesifikasi	Keterangan
<i>Baudrate</i>	9600,19200,28800,38400,57600(<i>default is 57600</i>)
<i>Working temperature rating</i>	-20 C + 50 C
<i>Woerking Humidity</i>	40% - 85 % RH
<i>Full Dimensions</i>	56 x 20 x 21.5 mm
<i>Exposed Dimension</i>	21 mm x 21 mm x 21 mm <i>triangular</i>
<i>Weight</i>	20 Ram

2.3 Sensor Dioda Laser [3]

Perangkat Semikonduktor yang mirip dengan dioda pemancar cahaya di mana sinar laser dibuat di persimpangan dioda. Yang menghasilkan radiasi koheren (semua gelombang berada pada frekuensi dan fase yang sama) dalam spektrum yang terlihat atau Inframerah (IR) ketika arus melewatinya.

Berdasarkan cara kerja, dioda laser dibedakan menjadi dua jenis, yaitu *Injecton laser diode* (ILD) dan Optically Pumped Semiconductor Laser. *Injecton laser diode* (ILD) memiliki kemiripan layaknya saudara kembar dengan LED. Keduanya dibuat berdasarkan proses dan teknologi yang hampir sama. Perbedaan yang tampak jelas pada dioda laser yaitu kehadiran sebuah saluran atau kanal panjang yang sempit dengan ujung yang reflektif. Kanal tersebut memiliki fungsi sebagai penuntun gelombang pada cahaya. Sebutan yang sering disemat oleh kanal adalah *Waveguide*.

Pengoperasian *Injecton laser diode* arus mengalir melalui persimpangan PN (*PN Junction*) dan menghasilkan cahaya seperti LED. Pancaran Fotonnya (*Photon*) disebabkan oleh bergabungnya elektro dan lubang (*holes*) di daerah persimpangan PN. Namun cahaya yang dihasilkan hanya dibatasi di dalam *waveguide* (penuntun cahaya) pada dioda laser. Di *waveguide*, cahaya laser direfleksikan dan kemudian diperkuat, lantas menghasilkan emisi terstimulasi sebelum dipancar keluar. Bentuk fisik dari modul sensor reflektif inframerah ini dapat dilihat pada Gambar 2.7.



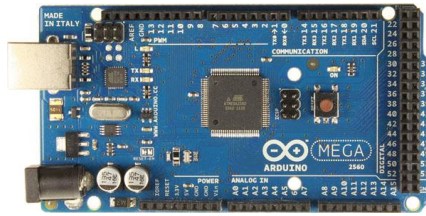
Gambar 2.7 Bentuk Fisik Modul Sensor Dioda Laser [3]

2.4 Arduino Mega 2560 [4]

Arduino Mega 2560 adalah *board* berbasis mikrokontroler pada ATmega 2560. *Board* ini memiliki pin I/O sejumlah 54 buah digital I/O pin (15 pin diantaranya adalah PWM), 16 pin analog *input*, 4 pin UART (*serial port hardware*). Arduino Mega 2560 dilengkapi dengan sebuah *oscillator* 16 Mhz, sebuah *port* USB, *power jack* DC, ICSP *header*, dan tombol *reset*. Pin – pin ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau sumber tekanan bisa didapat dari adaptor AC – DC atau baterai untuk menggunakannya. Untuk spesifikasi dari Arduino Mega 2560 ini dapat dilihat pada Tabel 2.2 dan untuk bentuk fisik dari Arduino Mega 2560 ini dapat dilihat pada Gambar 2.8.

Tabel 2.2 Spesifikasi Arduino Mega

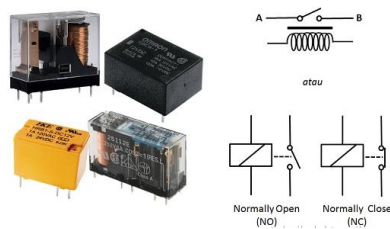
Spesifikasi	Keterangan
<i>Microcontroller</i>	ATmega 2560
<i>Operating Voltage</i>	5V
<i>Input Voltage (recommended)</i>	7 – 12 Volt
<i>Input Voltage (limits)</i>	6 – 20 Volt
<i>Digital I/O pins</i>	54 (<i>of which 14 provide PWM output</i>)
<i>Analog Input pins</i>	16
<i>DC Current per I/O Pin</i>	40 mA
<i>DC Current for 3.3V Pin</i>	50 mA
<i>Flash Memory</i>	256 KB <i>of which 8 KB used by bootloader</i>
<i>SRAM</i>	8 KB
<i>EEPROM</i>	4 KB
<i>Clock Speed</i>	16 MHz



Gambar 2.8 Arduino Mega [4]

2.5 *Relay* [5]

Relay adalah komponen elektronika yang berupa saklar atau *switch* elektrik yang dioperasikan menggunakan listrik. *Relay* juga biasa disebut sebagai komponen elektromekanikal yang terdiri dari dua bagian utama yaitu *coil* atau elektromagnet dan kontak saklar atau mekanikal. Komponen *relay* menggunakan prinsip elektromagnetik sebagai penggerak kontak saklar, sehingga dengan menggunakan arus listrik yang kecil atau *low power*, dapat menghantarkan arus listrik yang memiliki tegangan lebih tinggi. Berikut merupakan bentuk fisik dari *Relay* dan simbolnya dapat dilihat pada Gambar 2.9.



Gambar 2.9 Bentuk Fisik dan Simbol *Relay* [5]

Cara kerja *relay* yaitu apabila kumparan *coil* diberikan arus listrik, maka akan timbul gaya elektromagnet yang kemudian menarik armature untuk berpindah dari posisi sebelumnya (NC) ke posisi baru (NO) sehingga menjadi saklar yang dapat menghantarkan arus listrik di posisi barunya (NO). Posisi di mana armature tersebut berada sebelumnya (NC) akan menjadi *OPEN* atau tidak terhubung. Pada saat tidak dialiri arus listrik, armature akan kembali lagi ke posisi Awal (NC). *Coil* yang digunakan oleh *relay* untuk menarik *contact* poin ke

posisi *close* pada umumnya hanya membutuhkan arus listrik yang relatif kecil.

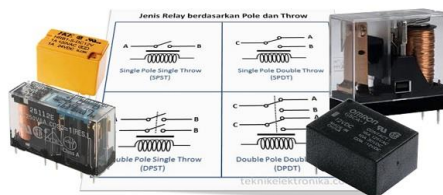
2.5.1 Jenis-jenis Relay

Relay merupakan salah satu jenis dari saklar maka istilah *pole* dan *throw* yang dipakai pada saklar juga berlaku pada *relay*. Berdasarkan penggolongan jumlah *pole* dan *throw* nya, sebuah *relay* dapat digolongkan menjadi :

- *Single Pole Single Throw (SPST)* : *Relay* golongan ini memiliki 4 terminal, 2 terminal untuk saklar dan 2 terminal lagi untuk *coil*.
- *Single Pole Double Throw (SPDT)* : *Relay* golongan ini memiliki 5 terminal, 3 terminal untuk saklar dan 2 terminalnya lagi untuk *coil*.
- *Double Pole Single Throw (DPST)* : *Relay* golongan ini memiliki 6 terminal, 4 terminal yang terdiri dari 2 pasang terminal untuk saklar dan 2 terminal lainnya lagi untuk *coil*. *Relay DPST* dapat dijadikan 2 saklar yang dikendalikan oleh 1 *coil*.
- *Double Pole Double Throw (DPDT)* : *Relay* golongan ini memiliki terminal sebanyak 8 buah, diantaranya 6 terminal yang merupakan 2 pasang *Relay SPDT* yang dikendalikan oleh 1 *coil*, sedangkan 2 terminal lainnya untuk *coil*.

Selain dari beberapa golongan *Relay* yang telah disebutkan, terdapat juga *relay* yang *pole* dan *throw*-nya lebih dari dua. Sehingga dapat memanfaatkan banyak kontak *relay* sesuai dengan kebutuhan.

Untuk melihat penggolongan *relay* berdasarkan jumlah *pole* dan *throw* dapat dilihat pada Gambar 2.10.



Gambar 2.10 Jenis Relay Berdasarkan Pole dan Throw [5]

2.5.2 Modul Relay Arduino

Saat ini berkembang pula modul *relay* Arduino yang praktis. Untuk fungsi dan cara kerjanya sama dengan *relay* biasanya yaitu sebagai saklar elektrik. Modul *relay* ini menggunakan sumber 5 Volt yang berasal dari Arduino. Modul *relay* ini termasuk *relay* SPDT (*Single Pole Double Throw*). Untuk bentuk fisik dari modul *relay* Arduino 8 *channel* ada pada Gambar 2.11 dan untuk modul *relay* 2 *channel* pada Gambar 2.12.



Gambar 2. 11 Modul Relay Arduino 8 Channel [5]



Gambar 2.12 Modul Relay Arduino 2 Channel [5]

Modul *Relay* Arduino yang digunakan pada *relay* 8 *channel* memiliki 8 *channel*. 10 pin di bagian bawah merupakan pin untuk Vcc 5 Volt, GND, dan IN 1-8. Dan untuk terminal merupakan terminal untuk COM di bagian tengah, dan kontak *Normally Open* dan *Normally Close*. Untuk spesifikasi dari modul *Relay* Arduino dapat dilihat pada Tabel 2.3

Tabel 2.3 Spesifikasi Modul Relay Arduino

Spesifikasi	Keterangan
Relay Maximum output	DC 30 Volt/10A, AC 250 Volt/10A.
8 Channel Relay Module with Opto-coupler.	
LOW Level Trigger expansion	

Spesifikasi	Keterangan
<i>board, which is compatible with Arduino control board</i>	
<i>Standard interface that can be controlled directly by microcontroller</i>	(8051, AVR, *PIC, DSP, ARM, ARM, MSP430, TTL logic).
<i>Relay of high quality low noise Relays SPDT.</i>	<i>A common terminal, a normally open, one normally closed terminal</i>
<i>Opto-Coupler isolation, for high voltage safety and prevent ground loop with microcontroller.</i>	
<i>Module Board</i>	138 mm x 56 mm

2.6 Motor DC [6]

Motor listrik merupakan sebuah perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Motor DC adalah motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Motor DC ini dapat berputar dengan arah yang berbeda yaitu berputar searah jarum jam dan sebaliknya. Motor DC yang digunakan memiliki spesifikasi yang dapat dilihat pada Tabel 2.4. Untuk bentuk fisik dari motor DC dapat dilihat pada Gambar 2.13



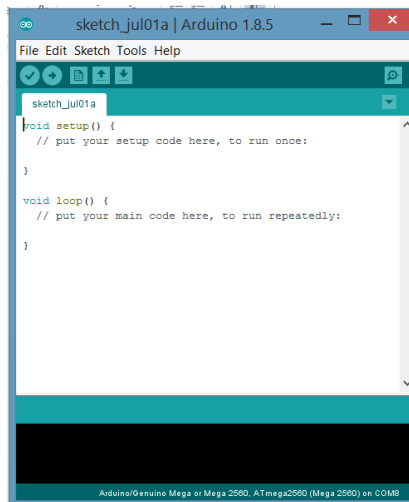
Gambar 2.13 Motor DC [6]

Tabel 2.4 Spesifikasi Motor DC

Spesifikasi	Keterangan
Tegangan	12-21,6 Volt
Arus	0,6 Ampere
Dimensi	90 x 50 x 60 mm
Kecepatan	70 Pm

lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang diberikan melalui sintaks pemrograman.

Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Untuk tampilan dari *sketch Software* Arduino IDE ini dapat dilihat pada Gambar 2.15.

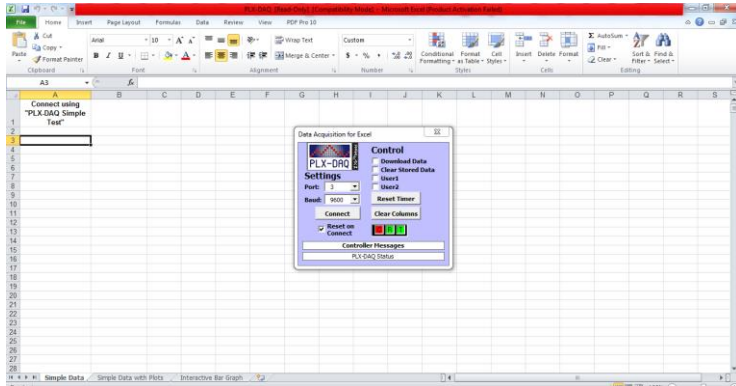


Gambar 2.15 *Sketch* dari *Software* Arduino IDE

2.9 *Software* PLX DAQ [9]

PLX-DAQ adalah *free software* yang digunakan penulis untuk mencatat data serial yang dikirim oleh arduino mega ke komputer (Parallax, 2012). Berikut contoh format data serial yang dikirimkan agar PLX-DAQ mencatat data di *microsoft excel* sesuai keinginan. Untuk memakainya pertama-tama harus menentukan *baudrate* yang akan digunakan, *baudrate* dan *port* yang digunakan ini harus dicocokkan dengan *baudrate* pada Arduino Mega, sebagai contoh jika pada arduino mega dikonfigurasi *baudrate* dengan nilai 9600 dan *port* 3, maka pada *software* PLX-DAQ juga harus menggunakan *baudrate*

yang sama. Untuk itu tampilan pada PLX DAQ dapat dilihat pada Gambar 2.16.

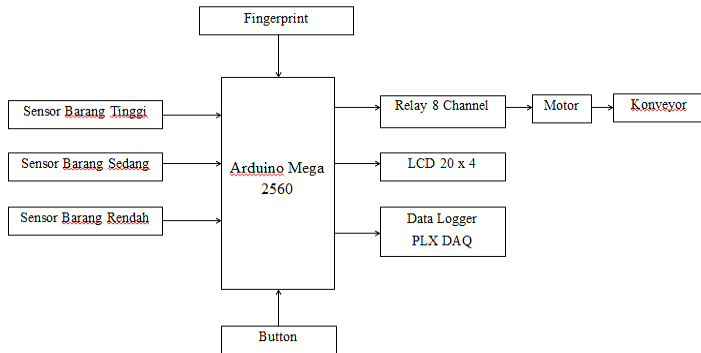


Gambar 2.16 Tampilan pada PLX DAQ

BAB III

PERANCANGAN *HARDWARE* DAN *SOFTWARE*

Bab ini membahas perancangan sensor *fingerprint* untuk mengaktifkan dan mematikan mesin konveyor serta perhitungan jumlah barang menggunakan mikrokontroler yang meliputi perancangan *hardware* dan *software*. Berikut ini merupakan blok fungsional diagram sistem secara keseluruhan.



Gambar 3.1 Diagram Fungsional Cara Kerja Konveyor

Dari Gambar 3.1 dapat dilihat bahwa pada alat ini menggunakan sensor *fingerprint* untuk mengaktifkan mesin konveyor. Serta terdapat *button* untuk menampilkan pilihan menu. Selain itu terdapat tiga sensor inframerah yang disusun secara *vertical* keatas dengan ketinggian yang berbeda. Untuk sensor barang rendah dipasang pada ketinggian 4 cm. Untuk sensor barang sedang dipasang pada ketinggian 5 cm. Dan untuk sensor barang tinggi dipasang pada ketinggian 6 cm. Ketika sensor *fingerprint* mendeteksi jari, serta *button* untuk menampilkan menu, dan ketika sensor inframerah mendeteksi barang. Maka data tersebut dikirimkan ke Arduino Mega untuk diolah. Arduino akan menerjemahkan hasil deteksi sensor *fingerprint* untuk mengaktifkan mesin konveyor serta menerjemahkan hasil deteksi sensor inframerah termasuk barang tinggi, sedang ataupun rendah. Sekaligus menghitungnya. Setelah diolah oleh Arduino Mega, data tersebut dikirimkan ke *relay*, LCD 20x4, dan PLX DAQ. Di PLX DAQ data berupa tanggal, waktu saat mengaktifkan dan mematikan

mesin konveyor, nama dan ID dari sidik jari yang telah terdaftar. Selain itu jumlah dari masing-masing barang juga ditampilkan pada PLX DAQ. Sedangkan di LCD 20x4 akan menampilkan pilihan menu untuk mendaftarkan sidik jari baru. Dan di *relay* akan mengontak ke motor DC dan motor tersebut akan menggerakkan Konveyor bagian atas dan bawah, *lift*, *pusher*. Masing-masing konveyor dikendalikan oleh satu *relay* dan satu motor, sedangkan untuk *lift* dan *pusher* dikendalikan oleh dua *relay* dan satu motor.

3.1 Perancangan *Hardware*

Perancangan *hardware* pada Tugas Akhir ini meliputi :

1. Perancangan Konveyor
2. Perancangan Contoh Barang
3. Perancangan Sensor *Fingerprint*
4. Perancangan Penggunaan Sensor Dioda Laser
5. Perancangan Penggunaan LCD 20x4
6. Perancangan Penggunaan *Relay*
7. Perancangan Tombol *Push Button*
8. Perancangan *Panel Box*
9. Perancangan *Wiring* Motor DC
10. Perancangan Regulator Tegangan
11. Perancangan *Software*

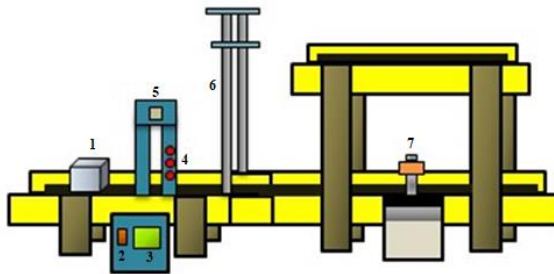
3.1.1 Perancangan Konveyor

Pada perancangan konveyor ini hanya perancangan untuk peletakan sensor-sensor yang telah kami modifikasi yang diletakkan pada konveyor. Karena konveyor ini keadaanya sudah jadi yang kami kembangkan dari Proyek Akhir milik Andrik Kurnia Pratama dan Rahmat Bagus Prasojodi tahun 2015, dengan judul Proyek Akhir “Mesin Pengelompok Barang Berdasarkan Ketinggian dan Konveyor Pengangkut Menggunakan PLC”. Begitu pula yang dikembangkan lagi dari judul sebelumnya yaitu milik Nuva Choironi Ersha tahun 2018, dengan judul Proyek Akhir Akhir “Monitoring Program Pengelompokan dan Penghitung Jumlah Barang pada Konveyor Menggunakan Mikrokontroler”.

Pada Proyek Akhir milik Nuva Choironi Ersha, untuk menyalakan konveyor masih dengan cara manual yaitu dengan tombol *on/off* dan untuk mengelompokkan barang berdasarkan ketinggian menggunakan sensor *infrared* serta mengelompokkan barang berdasarkan *barcode*. Sehingga kami mengembangkan dengan

menambahkan sensor *fingerprint* untuk aktivasi mesin konveyor dan untuk mengelompokkan barang berdasarkan ketinggian kami menggantinya dengan sensor dioda laser yang lebih tepat hasil deteksinya dibandingkan dengan sensor infrared. Serta mengelompokkan barang berdasarkan QR Code yang dideteksi menggunakan GM65 QR Code Scanner.

Untuk itu gambar rancangan peletakan sensor pada konveyor yang dibuat dapat dilihat pada Gambar 3.2



Gambar 3.2 Rancangan Konveyor

Keterangan bagian-bagian gambar :

1. Model Barang
2. Sensor *Fingerprint*
3. Layar LCD
4. Sensor Dioda Laser
5. GM65 QR Code Scanner
6. *Lift*
7. *Pusher*

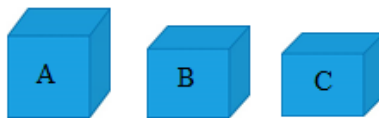
Dari rancangan konveyor tersebut dapat diketahui bahwa Sensor *Fingerprint* ini digunakan untuk mengaktifkan mesin konveyor. Di mana meliputi konveyor 1, konveyor 2, konveyor 3, konveyor 4, *pusher* dan *lift*. Ketika konveyor sudah aktif barang akan berjalan melewati konveyor dan ketika sudah melewati sensor ketinggian maka akan muncul jenis barang serta jumlah barang di LCD dan barang akan dikelompokkan sesuai dengan jenis barang.

3.1.2 Perancangan Contoh Barang

Pada Proyek Akhir ini menggunakan tiga macam contoh barang sebagai indikator ketinggian. Contoh barang ini terbuat dari kertas *duplex* yang dilapisi dengan kertas *concord* berwarna biru. Contoh barang yang digunakan berbentuk balok. Dan untuk ukuran dari barang yang digunakan berbeda-beda. Untuk ukuran dimensi dari contoh barang yang dibuat dapat dilihat pada Tabel 3.1. Untuk gambar penampang dari contoh barang dapat dilihat pada Gambar 3.3.

Tabel 3.1 Ukuran dari Contoh Barang

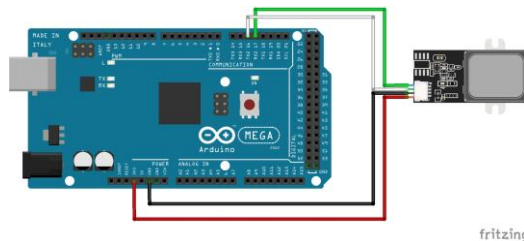
Jenis Barang	Panjang (cm)	Lebar (cm)	Tinggi (cm)
Tinggi	5	5	6
Sedang	5	5	5
Rendah	5	5	4



Gambar 3.3 Penampang dari Contoh Barang

3.1.3 Wiring Fingerprint

Pada Proyek Akhir Sensor Utamanya yaitu sensor *Fingerprint*. Sensor *Fingerprint* ini berfungsi untuk mengaktifkan dan mematikan mesin konveyor. Jadi ketika sidik jari telah terdeteksi maka mesin akan aktif (berjalan) dan untuk mematikannya yaitu dengan menempelkan sidik jari yang sama. Tetapi jika sidik jari tidak terdaftar maka konveyor tidak akan berjalan. Dan untuk *wiring Fingerprint* ke Arduino dapat dilihat pada Tabel 3.2



Gambar 3.4 Wiring Fritzing *Fingerprint* ke Arduino Mega

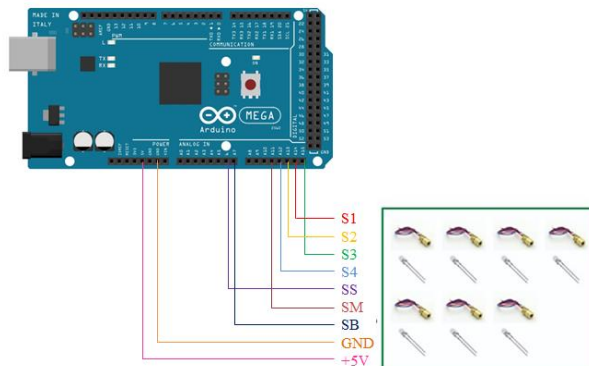
Untuk mengetahui lebih jelas mengenai *wiring fritzing* dari *Fingerprint* dapat dilihat pada Tabel 3.2

Tabel 3.2 Keterangan *Wiring Fritzing* Fingerprint ke Arduino mega

Pin Arduino Mega	Pin <i>Fingerprint</i>
3,3 Volt	Vcc
GND	GND
16	TX2
17	RX2

3.1.4 *Wiring Photo Dioda*

Pada Proyek Akhir yaitu menggunakan komponen *photo dioda*. Sensor ini berfungsi untuk mendeteksi barang yang melewati konveyor. Dan pada Proyek Akhir ini menggunakan 7 buah sensor yang sama. S1 yaitu terletak pada konveyor 1, S2 terletak pada konveyor 2, S3 terletak pada konveyor 3, S4 terletak pada konveyor 4. Sedangkan 3 *photo dioda* lainnya ini digunakan untuk mendeteksi barang berdasarkan ketinggian. Apabila SS (Sensor paling bawah) aktif maka sensor mendeteksi barang kecil. Apabila SM (Sensor paling bawah dan tengah) aktif maka sensor mendeteksi barang sedang. Dan apabila SB (sensor ketiganya) aktif maka sensor mendeteksi barang besar. Dan untuk *wiring* sensor ke Arduino Mega dapat dilihat pada Gambar 3.5



Gambar 3.5 *Wiring Fritzing Photo Dioda* ke Arduino Mega

Untuk mengetahui lebih jelas mengenai *wiring* fritzing dari *Photo Dioda* ke *Arduino* dapat dilihat pada Tabel 3.3

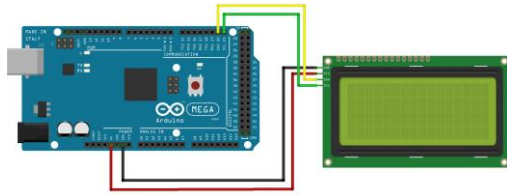
Tabel 3.3 Keterangan *Wiring* Fritzing *Photo Dioda* ke *Arduino* Mega

<i>Photo Dioda</i>	Pin <i>Arduino</i> Mega
S3	A15
S1	A14
S2	A13
S4	A12
SM	A11
SS	A7
SB	A6

3.1.5 *Wiring* LCD 20x4

LCD merupakan suatu perangkat elektronik yang berfungsi untuk menampilkan suatu data, huruf ataupun grafik. LCD yang digunakan pada perancangan di Proyek Akhir ini menggunakan LCD 20 x 4. Di mana LCD ini memiliki 20 kolom dan memiliki 4 baris, yang digunakan untuk menampilkan suatu huruf, angka, simbol, dll. LCD 20 x 4 ini digunakan untuk menampilkan data jumlah barang besar, jumlah barang sedang dan jumlah barang rendah. Selain itu juga untuk menampilkan total jumlah barang seluruhnya. Dan juga menampilkan menu-menu yang digunakan untuk mendaftarkan sidik jari baru pada *fingerprint*.

LCD ini membutuhkan tegangan kerja sebesar 5 Volt. Sedangkan untuk *Arduino* membutuhkan tegangan sebesar 9 Volt. Disini kami tidak menggunakan jalur paralel untuk *wiring* LCD ke *arduino* karena memakan banyak pin di *arduino* sehingga kami mengubah jalur paralel dari LCD ke I2C menggunakan modul I2C Converter supaya hanya membutuhkan sedikit pin. Modul LCD ini dikendalikan oleh *Arduino* Mega dan untuk *wiring* dari modul LCD 20x4 ini dihubungkan dengan I2C kemudian dari I2C dihubungkan ke *Arduino* Mega. Untuk *wiring*nya dapat dilihat pada Gambar 3.6.



Gambar 3.6 Wiring Fritzing LCD 20x4 ke Arduino Mega

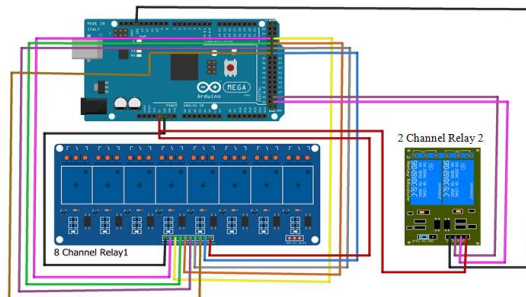
Untuk lebih jelas dalam mengetahui *wiring fritzing* dari LCD 20x4 dan Arduino Mega ini dapat dilihat pada Tabel 3.4

Tabel 3.4 Keterangan *Wiring Fritzing* dari LCD 20x4 ke Arduino

Pin Arduino Mega	Pin LCD 20 x 4
5 Volt	Vcc
GND	GND
20	SDA
21	SCL

3.1.6 *Wiring Modul Relay*

Relay dalam kerja sistem ini berfungsi untuk mengatur on/off motor DC penggerak konveyor. Kerja *relay* ini diatur oleh program Arduino Mega 2560. *Relay* yang digunakan merupakan modul *relay* Arduino dan untuk menggunakannya, cukup menyambungkan dari pin terminalnya.. Dan untuk *wiring fritzing* modul *relay* dengan Arduino Mega 2560 dapat dilihat pada Gambar 3.7



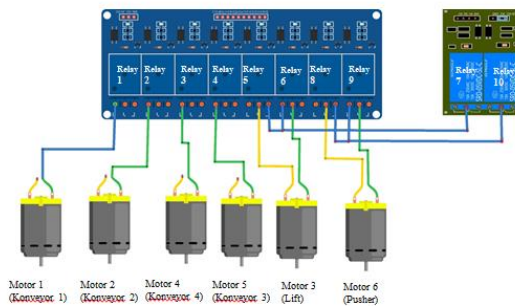
Gambar 3.7 Wiring Fritzing Arduino Mega ke Relay

Dan untuk lebih jelasnya mengenai *wiring fritzing* dari *relay* ke arduino dapat dilihat pada Tabel 3.5.

Tabel 3.5 Keterangan *Wiring Fritzing* dari Arduino Mega ke *Relay*

Pin Arduino	Pin Relay
5 Volt	Vcc (Relay 8 Channel)
Gnd	Gnd (Relay 8 Channel)
26	IN1 (Relay 8 Channel)
27	IN2 (Relay 8 Channel)
28	IN3 (Relay 8 Channel)
29	IN4 (Relay 8 Channel)
30	IN5 (Relay 8 Channel)
31	IN6 (Relay 8 Channel)
32	IN7 (Relay 8 Channel)
33	IN8 (Relay 8 Channel)
5 Volt	Vcc (Relay 2 Channel)
Gnd	Gnd (Relay 2 Channel)
51	IN1 (Relay 2 Channel)
53	IN2(Relay 2 Channel)

Modul *relay* Arduino juga terhubung dengan motor yang ada pada konveyor, karena *relay* berfungsi untuk mengontak nyala dan matinya motor. *Relay* ini diberikan tegangan sebesar 5 Volt. Untuk *wiring* modul *relay* ke motor dapat dilihat Gambar 3.8.



Gambar 3.8 *Wiring Fritzing* Modul *Relay* ke Motor

Untuk lebih jelasnya mengenai *wiring fritzing relay* ke motor dapat dijelaskan pada Tabel 3.6, Tabel 3.7, dan Tabel 3.8

Tabel 3.6 Keterangan *Wiring Fritzing Modul Relay* ke Motor

Pin Modul Relay	Motor	Keterangan
NO Relay 1	Motor 1 (+)	Menggerakkan Konveyor 1
NO Relay 2	Motor 2 (+)	Menggerakkan Konveyor 2
NO Relay 3	Motor 4 (+)	Menggerakkan Konveyor 4
NO Relay 4	Motor 5 (+)	Menggerakkan Konveyor 3

Tabel 3.7 Keterangan *Wiring Fritzing Relay* ke Motor untuk Lift

Pin Modul Relay	Motor (+)	Keterangan		
		Naik	Turun	Stop
COM Relay 5	3	High	Low	X
COM Relay 6	3	High	Low	X
COM Relay 7	3	Low	Low	High

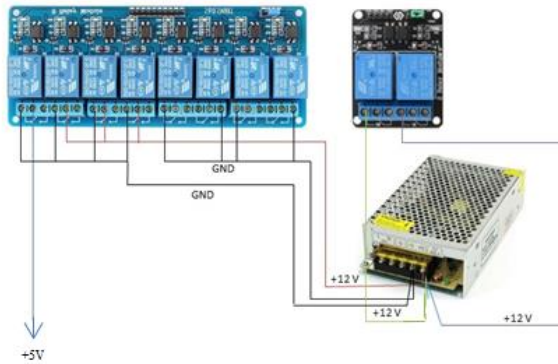
Berdasarkan Tabel 3.7 *Relay 7* yaitu berfungsi sebagai *enable (active low)*. Sedangkan *Relay 5* dan *6* berfungsi sebagai *direction*. Di mana *direction* ini merupakan arah putaran motor. Ketika *COM Relay 5* bernilai *high* maka akan mendapatkan tegangan 12 volt, sedangkan ketika *relay 6* bernilai *high* maka akan tersambung ke *ground* sehingga arah putaran motor untuk *lift* dengan kondisi *CW (Clock Wise)*. Di mana kondisi *CW* yaitu arah putaran motor searah jarum jam sehingga *lift* naik keatas. Kemudian ketika *COM relay 5* bernilai *low* maka akan tersambung ke *ground*. Sedangkan *COM relay 6* ketika bernilai *low* akan mendapat tegangan 12 volt sehingga arah putaran untuk *lift* dengan kondisi *CCW(Counter Clock Wise)*. Di mana kondisi *CCW* yaitu arah putaran motor berlawanan arah jarum jam sehingga *lift* turun ke bawah. Disini *lift* memiliki kondisi *stop* ketika *enable* atau *relay 7* bernilai *High*. Saat kondisi *relay 7* *high* maka *relay 5* dan *6* yang berfungsi sebagai *direction* ini tidak dipedulikan lagi (*don't care*). Sedangkan untuk keterangan *wiring relay* ke motor untuk menggerakkan *pusher* dapat dilihat pada tabel 3.8.

Tabel 3. 8 Keterangan *Wiring Fritzing Relay* ke Motor untuk *Pusher*

Pin Modul Relay	Motor (+)	Keterangan		
		Dorong	Tarik	Stop
COM Relay 8	6	High	Low	X
COM Relay 9	6	High	Low	X
COM Relay 10	6	Low	Low	High

Berdasarkan Tabel 3.8 untuk menggerakkan *pusher* agar bisa berputar dua arah yaitu menggunakan tiga *relay* yang berbeda. Disini menggunakan 2 dari Modul *Relay 8 channel* dan 1 dari Modul *Relay 2 channel* yang 5V. Di mana *Relay 10* yaitu berfungsi sebagai *enable* (*active low*). Sedangkan *Relay 8* dan *9* berfungsi sebagai *direction*. Di mana *direction* ini merupakan arah putaran motor. Ketika COM *Relay 8* bernilai *high* maka akan mendapatkan tegangan 12 volt, sedangkan ketika *relay 9* bernilai *high* maka akan tersambung ke *ground* sehingga arah putaran motor untuk *lift* dengan kondisi CW (*Clock Wise*). Di mana kondisi CW yaitu arah putaran motor searah jarum jam sehingga *pusher* bergerak mendorong. Kemudian ketika COM *relay 8* bernilai *low* maka akan tersambung ke *ground*. Sedangkan COM *relay 9* ketika bernilai *low* akan mendapat tegangan 12 volt sehingga arah putaran untuk *pusher* dengan kondisi CCW(*Counter Clock Wise*). Di mana kondisi CCW yaitu arah putaran motor berlawanan arah jarum jam sehingga *pusher* bergerak mundur. Disini *pusher* memiliki kondisi *stop* ketika *enable* atau *relay 10* bernilai *High*. Saat kondisi *relay 10* *high* maka *relay 8* dan *9* yang berfungsi sebagai *direction* ini tidak dipedulikan lagi (*don't care*).

Selain tersambung dengan Arduino dan Motor DC, Modul *Relay* juga tersambung ke *power supply*. *Power supply* pada alat ini sebagai sumber untuk menggerakkan motor di konveyor. *Power supply* merubah tegangan 220 Volt AC menjadi 12 Volt DC. Untuk *wiring fritzing relay* ke *power supply* dapat dilihat pada Gambar 3.9.



Gambar 3.9 Wiring Fritzing Power Supply ke Modul Relay

Untuk lebih jelasnya mengenai *wiring fritzing power supply* ke modul *relay* dapat dilihat pada Tabel 3.9

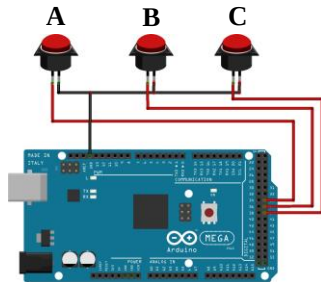
Tabel 3.9 Keterangan Wiring Fritzing Relay ke Power supply

Power supply	Modul Relay
+12 Volt	COM Relay 2
	COM Relay 3
	COM Relay 4
	NO Relay 7 (2 Channel)
	NO Relay 10 (2 Channel)
+5 Volt	COM Relay 1
GND	NO Relay 1
	NO Relay 2
	NO Relay 5 dan NC Relay 6 (8 Channel)
	NO Relay 3
	NO Relay 4
	NO Relay 8 dan NC Relay 9 (8 Channel)

3.1.7 Wiring Push Button ke Arduino

Push button pada proyek akhir ini digunakan untuk memilih menu pada LCD. Disini kami menggunakan 3 *Push button*. *Push button* paling kiri (*push button A*) digunakan untuk mengecek kondisi semua sensor yang ditampilkan pada LCD dan untuk memilih NIP saat pendaftaran sidik jari baru. *Push button* tengah (*push button B*) digunakan untuk tombol kembali ke menu awal. Sedangkan *push*

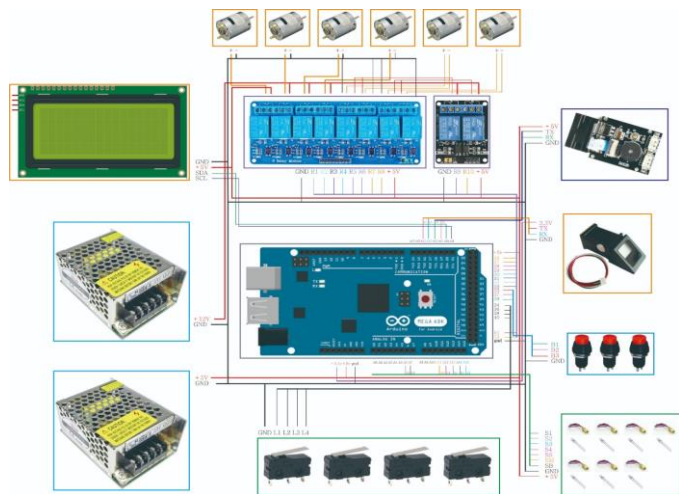
button paling kanan (*push button C*) digunakan untuk mendaftarkan sidik jari baru. Berikut ini *wiring fritzing* dari *push button* ke arduino dapat dilihat pada Gambar 3.10



Gambar 3.10 *Wiring Fritzing Push Button* ke Arduino

3.1.6 *Wiring Secara Keseluruhan*

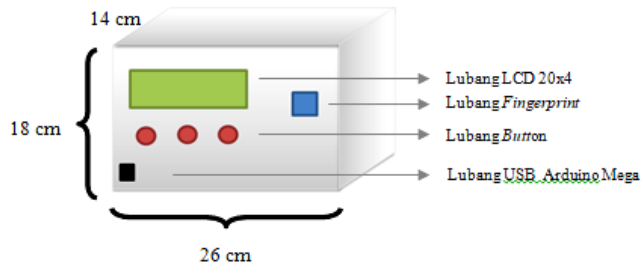
Setelah membuat perancangan *wiring* dari masing-masing komponen maka semua komponen dirangkai menjadi satu untuk menjalankan sistem ini, sehingga perancangan untuk *wiring* secara keseluruhan dapat dilihat pada Gambar 3.11



Gambar 3.11 *Wiring Secara Keseluruhan*

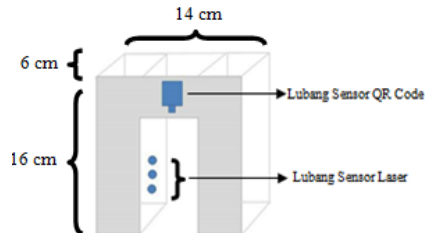
3.1.7 Perancangan *Panel Box*

Pada perancangan *panel box* ini digunakan sebagai tempat atau wadah bagi sensor *fingerprint*, LCD, *button* dan lubang USB arduino. Di mana *fingerprint* ini digunakan untuk mendeteksi sidik jari karyawan yang akan menyalakan mesin konveyor. Sedangkan LCD menampilkan jumlah barang yang terdeteksi. Kemudian *button* untuk pilihan menu dan penambahan sidik jari baru. *Panel box* ini terbuat dari *acrilyc* berwarna putih sebagai bahan utamanya. Untuk ukuran dari *panel box* ini adalah 26 cm x 14 cm x 18 cm. *Design* dari *panel box* ini dapat dilihat pada Gambar 3.12.



Gambar 3.12 *Design* dari *Panel Box Fingerprint, LCD, dan Button*

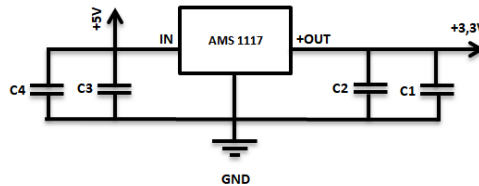
Sedangkan perancangan *panel box* ini digunakan sebagai tempat atau wadah bagi sensor dioda laser dan QR Code Scanner. Pada panel *box* ini terdapat 3 sensor dioda laser yang digunakan untuk mendeteksi barang berdasarkan ketinggian. *Panel box* ini terbuat dari *acrilyc* berwarna putih sebagai bahan utamanya. Untuk ukuran dari *panel box* ini adalah 14 cm x 6 cm x 16 cm. *Design* dari *panel box* ini dapat dilihat pada Gambar 3.13.



Gambar 3.13 *Design* dari *Panel Sensor Dioda Laser*

3.1.8 Perancangan Regulator Tegangan

Regulator Tegangan pada Proyek Akhir ini digunakan untuk mengatur tegangan yang masuk pada sensor *fingerprint*. Karena tegangan yang dibutuhkan untuk menyuplai sensor *fingerprint* yaitu sebesar +3,3 V Volt dan GND. Dan untuk gambar skematik dari rangkaian regulator tegangan 3,3 Volt dapat dilihat pada Gambar 3.14



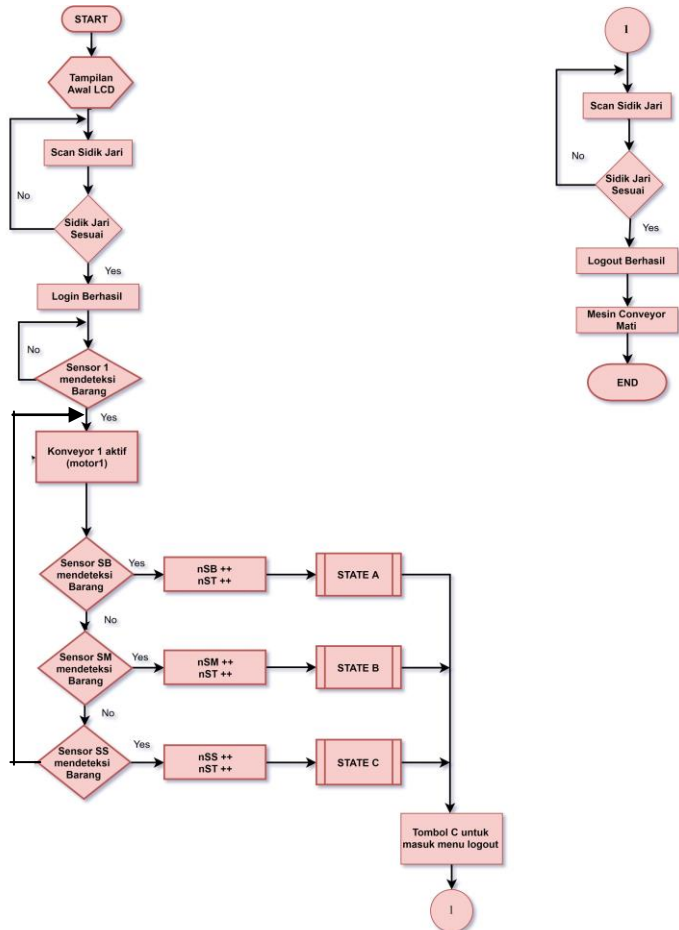
Gambar 3.14 Rangkaian Regulator Tegangan

Rangkaian regulator tegangan pada Gambar 3.14 merupakan penurun tegangan dari 5 Volt ke 3,3 Volt. Untuk menurunkan tegangan menggunakan IC Regulator /AMS 1117.

3.2 Perancangan Software

Pada perancangan perangkat lunak ini bertujuan untuk menunjang perangkat keras (*hardware*) agar terbentuk menjadi satu kesatuan sistem. Perangkat lunak yang digunakan pada sistem ini yaitu program Arduino Mega 2560 untuk membuat dan merencanakan program dalam bahasa C menggunakan Arduino IDE. Dalam perancangan *software* ini diperlukan beberapa tahapan yang harus dilakukan terlebih dahulu. Sebelum membuat program lebih baik membuat *Flowchart* terlebih dahulu agar program menjadi lebih sederhana dan sesuai dengan alur kerjanya. Untuk itu perlu membuat *Flowchart* untuk merancang *software*. Berikut ini merupakan perancangan *software* secara keseluruhan. Dimana alur kerjanya yaitu diawali dengan menyalakan tombol *power* untuk mengaktifkan LCD dan sensor. Setelah itu *login* terlebih dahulu supaya mesin konveyor dapat aktif (berjalan). Ketika sudah berhasil *login* maka ketika meletakkan barang pada ujung konveyor 1 dan menghalangi sensor 1 maka berjalan. Setelah itu barang berjalan melewati sensor dioda laser untuk dideteksi. Setelah terdeteksi maka barang akan dikelompokkan sesuai

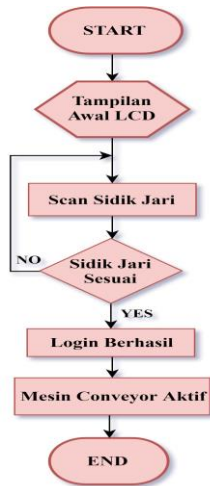
dengan jenis barangnya. Disini kami menggunakan 3 jenis barang yaitu kecil, sedang dan besar. Ketika proses pengelompokkan barang telah selesai maka untuk memamatkannya harus *logout* terlebih dahulu. Data selama proses berlangsung otomatis tersimpan pada PLX DAQ. Untuk itu *Flowchart* keseluruhan dapat dilihat pada Gambar 3.15.



Gambar 3.15 *Flowchart* Secara Keseluruhan

3.2.1 Flowchart Login untuk Aktivasi Mesin Konveyor

Untuk membuat program supaya sensor *fingerprint* ini dapat mengaktifkan mesin konveyor. Maka dari itu perlu membuat *Flowchart* terlebih dahulu supaya saat membuat program sensor *fingerprint* untuk mengaktifkan konveyor ini menjadi lebih mudah dan Untuk itu dibuatlah *Flowchart* seperti Gambar 3.16



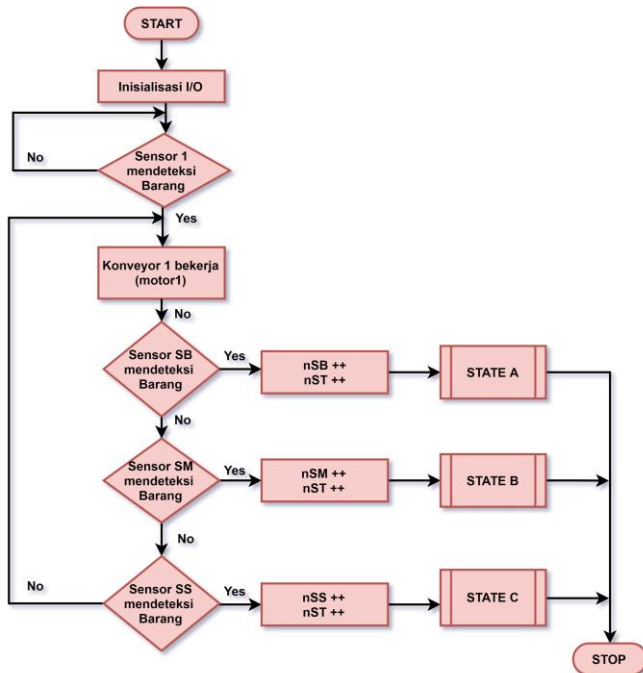
Gambar 3.16 Flowchart Login untuk Mengaktifkan Konveyor

Pada Gambar 3.16 merupakan *Flowchart login* untuk mengaktifkan mesin konveyor. Sebelum *login* maka perlu mendaftarkan sidik jari terlebih dahulu melalui program *enroll* seperti pada Gambar 4.1. Setelah jari terdaftar pada sensor *fingerprint* maka dapat melakukan *login* yang prosesnya seperti pada Gambar 3.15. Jadi, seorang karyawan atau operator dapat mengaktifkan mesin jika *login* sudah berhasil. Awalnya pada konveyor terdapat LCD yang tampilan awalnya yaitu “Place Your Finger to Login” yang berarti perintah untuk *login* dengan menempelkan sidik jari pada sensor *fingerprint*. Setelah menempelkan sidik jari pada sensor maka sensor *fingerprint* akan menscan (mendeteksi sidik jari). Ketika sidik jari sesuai maka akan muncul pemberitahuan pada LCD bahwa *Login* berhasil. Jika sidik jari tidak sesuai maka akan kembali ke proses awal yaitu untuk menscan sidik jari lagi hingga berhasil. Setelah *login* berhasil maka

mesin konveyor aktif. Dan setelah ada barang yang melewati sensor 1 (sensor dioda laser) yang terdapat pada ujung konveyor pertama maka konveyor akan berjalan untuk melakukan proses selanjutnya. Jika *login* tidak berhasil maka mesin konveyor tidak akan aktif.

3.2.2 *Flowchart* Pengelompokan Barang Berdasarkan Ketinggian

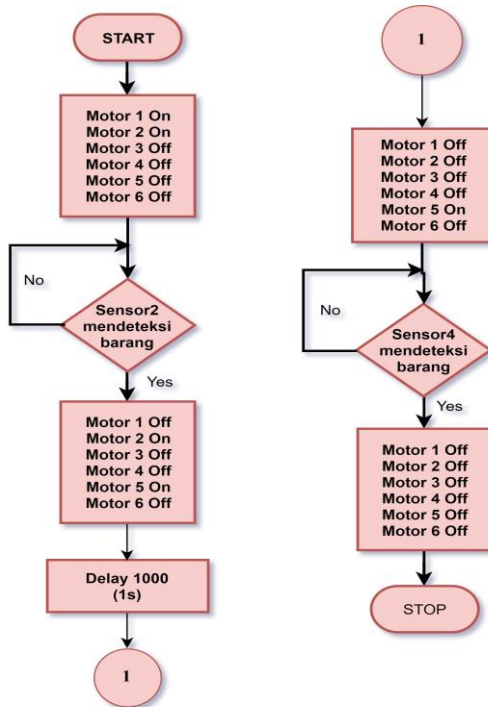
Untuk membuat program supaya sensor dioda laser yang disusun secara vertikal ini dapat mendeteksi barang berdasarkan ketinggian serta dapat menghitung jumlah barang secara otomatis yang ditampilkan pada LCD. Maka dari itu perlu membuat *Flowchart* terlebih dahulu supaya saat membuat program lebih mudah dan Untuk itu dibuatlah *Flowchart* seperti Gambar 3.17



Gambar 3.17 *Flowchart* Pengelompokan Barang

Pada Gambar 3.17 menunjukkan bahwa alur kerja untuk mengelompokkan barang berdasarkan ketinggian dengan menggunakan sensor dioda laser. Alur kerjanya yaitu diawali dengan *start/mulai* setelah itu inisialisasi *input/output*. Kemudian ketika sensor 1 (sensor yang berada di ujung konveyor 1) mendeteksi suatu barang maka motor 1 akan bekerja menggerakkan konveyor 1. Setelah konveyor 1 berjalan maka barang berjalan melewati sensor ketinggian atau sensor dioda laser yang disusun secara vertikal. Sensor ketinggian ini ada 3 macam SB, SM, dan SS. Di mana sensor SB ini merupakan sensor untuk mendeteksi barang besar. Sedangkan sensor SM yaitu sensor untuk mendeteksi barang sedang dan sensor SS yaitu sensor yang digunakan untuk mendeteksi barang kecil. Jadi, ketika Sensor SB mendeteksi suatu barang maka nSB++ dan nST++ yang artinya jumlah barang besar serta total barang besar yang ada pada LCD bertambah 1 dan melanjutkan ke proses STATE A. Ketika Sensor SM mendeteksi suatu barang maka nSM++ dan NST++ yang artinya jumlah serta total barang sedang yang ada pada LCD bertambah 1 dan melanjutkan ke proses STATE B. Ketika Sensor SS mendeteksi suatu barang maka nSS++ dan NST++ yang artinya jumlah serta total barang kecil yang ada pada LCD bertambah 1 dan melanjutkan ke proses STATE C. Untuk *Flowchart* STATE A, STATE B, dan STATE C dapat dilihat pada Gambar 3.18, 3.19 dan 3.20

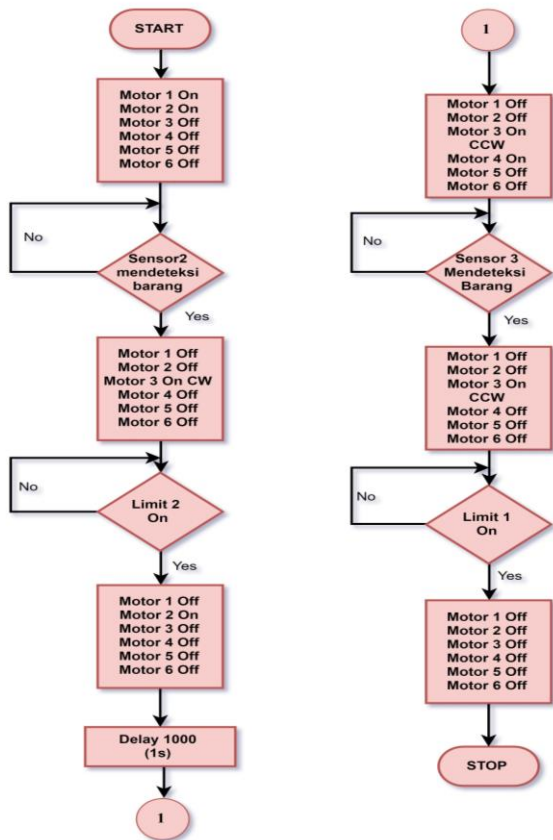
Flowchart STATE A, merupakan alur kerja dari pengelompokan barang besar. Dimana alur kerja yang kami inginkan yaitu setelah sensor SB mendeteksi barang besar maka konveyor 1 masih aktif dan kemudian motor 2 aktif untuk menggerakkan konveyor 2. Setelah barang berjalan di konveyor 2 maka sensor 2 akan mendeteksi barang. Ketika sensor 2 mendeteksi barang maka motor 1 mati dan konveyor 1 pun otomatis mati. Setelah itu motor 2 tetap aktif dan motor 5 aktif untuk menggerakkan konveyor 3 sehingga barang dapat berjalan menuju konveyor 3. Pada saat barang sudah berada di konveyor 3 maka selang 1 detik motor 2 mati sehingga konveyor 2 mati. Setelah itu barang melewati sensor 4 yang ada pada ujung konveyor 3. Ketika barang terdeteksi oleh sensor 4 maka motor 5 mati sehingga konveyor 3 pun berhenti dan barang terjatuh dari konveyor. Sehingga barang besar termasuk pengelompokkan dengan barang berjalan lurus. Untuk itu *Flowchart* STATE A dapat dilihat pada Gambar 3.18.



Gambar 3.18 Flowchart STATE A

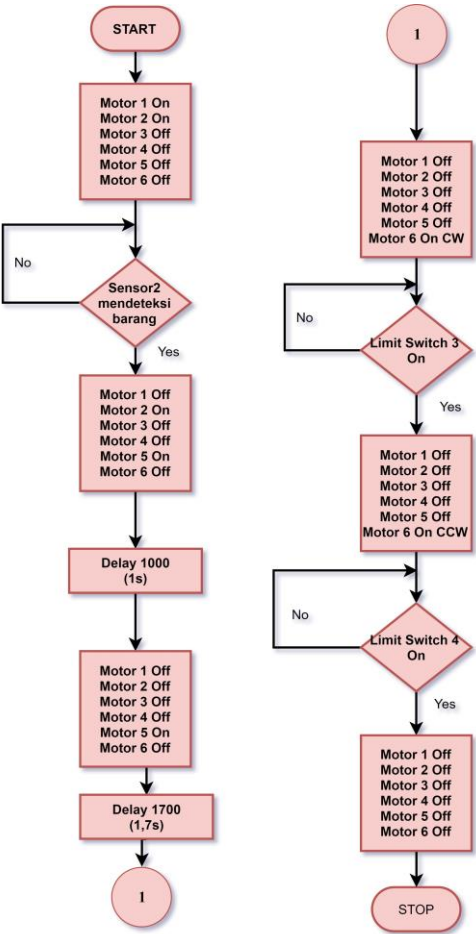
Selanjutnya yaitu *Flowchart* STATE B, dimana STATE B ini merupakan alur kerja dari pengelompokan barang sedang. Alur kerja yang kami inginkan yaitu setelah sensor SM mendeteksi barang sedang maka konveyor 1 masih aktif dan kemudian motor 2 aktif untuk menggerakkan konveyor 2. Setelah barang berjalan di konveyor 2 maka sensor 2 akan mendeteksi barang. Ketika sensor 2 mendeteksi barang maka motor 1 mati dan konveyor 1 pun otomatis berhenti. Setelah itu motor 2 tetap aktif dan motor 5 aktif untuk menggerakkan konveyor 3 sehingga barang dapat berjalan menuju konveyor 3. Pada saat barang sudah berada di konveyor 3 maka selang 1 detik motor 2 mati sehingga konveyor 2 berhenti. Setelah itu pada saat konveyor 3 berjalan selang 1,7 detik motor 5 mati sehingga konveyor 3 berhenti dan motor 6 akan berputar searah jarum jam/ CW (*Clock wise*). Di mana motor 6 ini untuk menggerakkan *pusher* sehingga jika motor

berputar searah jarum jam maka *pusher* akan bergerak maju/ mendorong barang tersebut hingga jatuh. Pada saat *pusher* bergerak maju maka *limit switch* 3 on sehingga motor 6 berputar berlawanan arah jarum jam/CCW (*Counter clock wise*). Maka *pusher* akan bergerak mundur dan *limit switch* 4 on. Ketika *limit switch* on maka motor 6 mati atau *pusher* berhenti. Jadi barang sedang tergolong barang yang terdorong keluar oleh *pusher*. Untuk itu *Flowchart* STATE B dapat dilihat pada Gambar 3.19



Gambar 3.19 Flowchart STATE B

Selanjutnya yaitu *Flowchart* STATE C, yang merupakan alur kerja dari pengelompokan barang kecil. *Flowchart* STATE C dapat dilihat pada Gambar 3.20

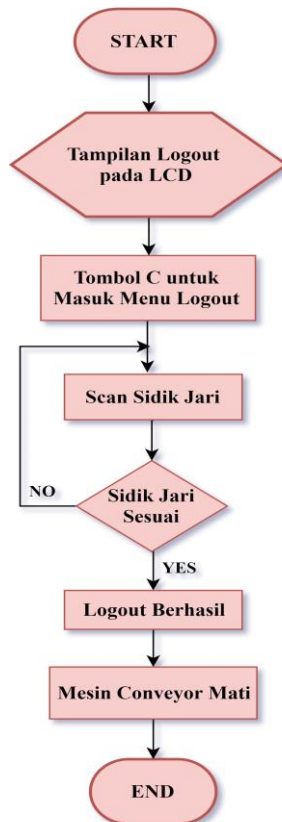


Gambar 3.20 *Flowchart* STATE C

Pada Gambar 3.20 alur kerjanya yaitu setelah sensor SS mendeteksi barang kecil maka konveyor 1 masih aktif dan kemudian motor 2 aktif untuk menggerakkan konveyor 2. Setelah barang berada di konveyor 2 maka sensor 2 akan mendeteksi barang. Ketika sensor 2 mendeteksi barang maka motor 1 dan motor 2 mati sehingga konveyor 1 dan 2 pun berhenti. Tetapi motor 3 aktif berputar searah jarum jam/ CW (*Clock wise*). Di mana motor 3 ini untuk menggerakkan *lift*. Jadi ketika motor berputar searah jarum jam maka *lift* bergerak naik keatas membawa barang. Setelah sampai atas maka *limit switch* 2 on dan motor 2 on sehingga konveyor 2 bergerak. Setelah itu selang 1 detik, maka motor 4 on sehingga konveyor 4 berjalan sehingga barang dari konveyor 2 sudah berjalan menuju konveyor 4. Selang 1 detik, maka motor 2 off dan motor 3 berputar berlawanan arah jarum jam / CCW (*Counter clock wise*) sehingga *lift* turun. Setelah itu barang yang berjalan di konveyor 4 terdeteksi oleh sensor 3, sehingga barang jatuh dan konveyor 4 berhenti. Pada saat itu *lift* masih bergerak turun dan sampai di bawah maka *limit switch* 1 on , sehingga *lift* berhenti. Jadi, barang kecil termasuk pengelompokkan untuk naik ke atas *lift*.

3.2.3 **Flowchart Logout untuk Mematikan Konveyor**

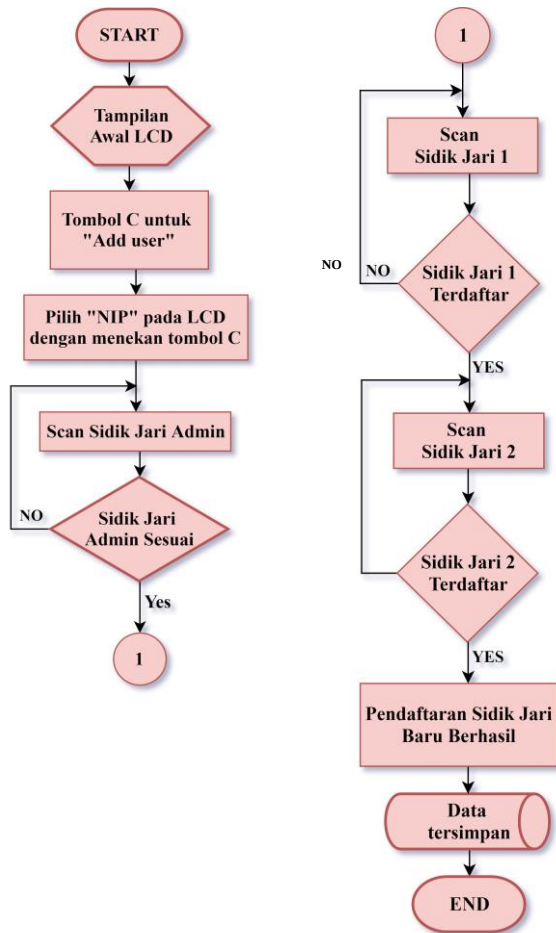
Alur kerja yang kami inginkan untuk mematikan mesin konveyor yaitu setelah melakukan perhitungan barang dan pengelompokkan barang secara otomatis, maka pada LCD terdapat tampilan *logout* yang digunakan untuk mematikan mesin konveyor. Setelah itu untuk memilih menu *logout* ini tekan tombol C. Jadi jika belum memilih menu *logout* maka tidak dapat mematikan mesin secara tiba-tiba disaat ditengah proses distribusi barang pada konveyor. Setelah itu ada perintah pada LCD “*Place your finger to logout*” yang berarti perintah meletakkan sidik jari pada sensor *fingerprint* untuk melakukan *logout*. Apabila sidik jari sesuai maka *logout* berhasil. Dan apabila sidik jari tidak sesuai maka akan kembali ke proses awal yaitu *scan* sidik jari kembali hingga berhasil. Setelah *logout* berhasil maka mesin konveyor mati sehingga tidak bisa berjalan. Untuk itu *Flowchart Logout* untuk mematikan mesin dapat dilihat pada Gambar 3.21



Gambar 3.21 *Flowchart Logout untuk Mematikan Konveyor*

3.2.4 *Flowchart Penambahan Sidik Jari Baru*

Pada sistem ini dirancang untuk bisa menambahkan sidik jari baru tanpa perlu mengubah program. Untuk itu dibuat *Flowchart* terlebih dahulu supaya lebih mudah untuk membuat program dan hasilnya dapat sesuai dengan yang diinginkan. *Flowchart* dapat dilihat pada Gambar 3.22



Gambar 3. 22 Flowchart Penambahan Sidik jari Baru

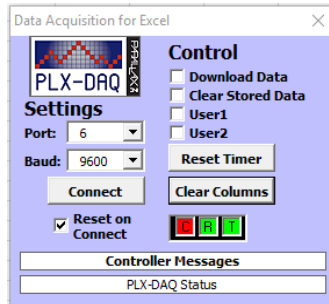
Berdasarkan Gambar 3.22 merupakan alaur kerja untuk menambahkan sidik jari baru tanpa perlu merubah program. Untuk menambahkan sidik jari baru hanya memilih menu pada LCD. Jadi, jika ada karyawan baru untuk menambahkan sidik jari baru tekan tombol *push button* C pada panel *box* seperti pada Gambar 4.13 untuk

memilih menu “*Add User*” yang ditampilkan pada LCD. Setelah itu akan muncul tampilan pilihan NIP, NIP dari 1110001-1110010. Setelah NIP dipilih maka tekan “*Next*” menggunakan tombol *push button* C. Kemudian akan muncul perintah untuk *scanning* sidik jari Admin terlebih dahulu. Karena saya *setting* untuk penambahan sidik jari baru yang mendaftarkan yaitu Admin, supaya NIP tidak sama dengan karyawan lain dan supaya tidak sembarang orang dapat mendaftarkan sidik jari baru. Setelah sensor *fingerprint* mendeteksi sidik jari admin dan sidik jari admin sesuai, maka akan masuk ke proses selanjutnya yaitu *scan* sidik jari baru 1. Jika sidik jari baru 1 berhasil terdaftar maka akan ke proses selanjutnya. Setelah itu *scan* sidik jari baru ke 2, setelah sidik jari baru 2 terdaftar maka proses pendaftaran sidik jari baru telah berhasil. Pada sistem ini kami *setting* setiap NIP nya untuk mendaftarkan 2 jari, karena jika ada salah satu sidik jari yang luka atau tidak bisa digunakan bisa menggunakan jari satunya yang sudah terdaftar.

3.2.5 Perancangan Pendataan Barang Menggunakan Software PLX DAQ

Untuk perancangan pendataan barang berdasarkan ketinggian yang telah melewati konveyor dapat tersimpan secara otomatis pada komputer yaitu dengan menggunakan *software* PLX DAQ. Karena dengan menggunakan *database* PLX DAQ maka semua data saat *login* dan *logout* menggunakan *fingerprint* serta jumlah barang akan tersimpan otomatis dalam bentuk Ms.Excel dan sesuai dengan waktu nyata. Jadi dengan disimpan pada *database* PLX DAQ ini, maka ketika pengguna akan melakukan pengecekan ulang data terdahulu dapat dilihat dalam *database* ini sendiri. Secara tidak langsung *database* ini merupakan suatu sarana untuk *membackup* keseluruhan data yang diukur agar data tersebut menjadi aman.

Di mana *software* PLX DAQ ini yaitu *software* untuk mencatat data serial yang dikirim oleh arduino mega 2560 ke computer dalam bentuk Ms.Excel. Untuk menggunakannya pertama buka PLX DAQ dan sesuaikan *port* serta *baudrate* yang terhubung dengan arduino mega 2560. Untuk itu dapat dilihat pada Gambar 3.23



Gambar 3.23 Connect PLX DAQ ke Arduino

Pada Gambar 3.23 menunjukkan bahwa *port* yang digunakan yaitu *port* 6 dan *baudrate* 9600 karena pada arduino mega 2560 dikonfigurasi dengan *baudrate* 9600 sehingga pada *software* PLX DAQ harus menggunakan *baudrate* yang sama. Selanjutnya untuk menampilkan data-data perlu untuk memrogram pada Arduino IDE terlebih dahulu. Pada Proyek Akhir ini kami merancang bahwa data yang ditampilkan berupa tanggal, waktu, status (*Login/Logout*), NIP, dan jumlah masing masing barang yang terdeteksi oleh *scanner* QR Code maupun terdeteksi oleh sensor ketinggian (dioda laser). Untuk menampilkan judul dari masing-masing kolom tersebut programnya yaitu sebagai berikut : `Serial.println("LABEL,Date,Time,Status,NIP,nQB,nQM,nQS,nQT,nSB,nSM,nSS,nST");` Berdasarkan program tersebut label digunakan untuk menentukan *header* pada kolom, untuk pindah ke kolom selanjutnya digunakan tanda koma. Maka didapatkan hasil perancangan kolom pada PLX DAQ seperti pada Gambar 3.24.

A	B	C	D	E	F	G	H	I	J	K	L
Date	Time	Status	NIP	nQB	nQM	nQS	nQT	nSB	nSM	nSS	nST
6/27/2019	8:36:51 PM	LOGIN	ADMIN	0	0	0	0	0	0	0	0
6/27/2019	8:39:58 PM	LOGOUT	ADMIN	0	0	0	0	1	1	1	3

Gambar 3.24 Perancangan Data pada *Software* PLX DAQ

Untuk penulisan data digunakan *DATA*, *TIME*, setelah itu dilanjutkan dengan *print serial* data yang diinginkan sesuai dengan *headernya*. Waktu akan selalu muncul didepan, baru setelah itu data-data yang lainnya. Untuk program menampilkan data pada PLX DAQ dapat dilihat pada Lampiran B.4.

Data akan tersimpan secara otomatis jika PLX DAQ sudah di *connect* kan. Setelah *disconnect*kan maka judul dari masing-masing kolom akan muncul. Setelah sidik jari terdeteksi saat proses *login* dan *logout* maka data akan tersimpan otomatis pada PLX DAQ seperti pada Gambar 3.24. Dimana jumlah barang saat *login* tersebut masih 0 karena barang belum ada yang melewati konveyor. Setelah *login* berhasil maka barang melewati konveyor yang dideteksi oleh *Qr Code scanner* dan sensor ketinggian (dioda laser) untuk dihitung secara otomatis yang ditampilkan pada LCD dan dikelompokkan. Setelah itu untuk mematikan mesin harus *logout* terlebih dahulu. Ketika *logout* maka data karyawan serta jumlah barang yang telah diselesaikan akan tersimpan secara otomatis pada PLX DAQ. Untuk jumlah barang pada Gambar 3.24 dapat dilihat pada kolom nQB = jumlah barang besar berdasarkan *QR Code*, nQM = jumlah barang sedang berdasarkan *QR Code*, nQS = jumlah barang kecil berdasarkan *QR Code*, nQT = total keseluruhan barang berdasarkan *QR Code*, nSB = jumlah barang besar berdasarkan sensor ketinggian, nSM = jumlah barang sedang berdasarkan sensor ketinggian, nSS = jumlah barang kecil berdasarkan sensor ketinggian, nST = total keseluruhan barang berdasarkan sensor ketinggian.

-----Halaman ini sengaja dikosongkan-----

BAB IV

PENGUJIAN DAN ANALISA

Untuk mengetahui apakah tujuan dari pembuatan sistem ini telah tercapai atau tidak, maka perlu diadakannya sebuah pengujian dan analisa terhadap alat yang telah dibuat. Dan sebagai acuan yang tidak terpisahkan adalah adanya proses evaluasi sehingga akan dapat dilakukan langkah-langkah positif guna membawa alat ini kearah yang lebih baik. Adapun pengujian yang dilakukan dengan dua tahap yaitu pengujian pada tiap sub sistem dan secara keseluruhan. Untuk pengujian sub sistem terdiri dari pengujian sensor *fingerprint*, pengujian sensor dioda laser, pengujian penambahan sidik jari baru, pengujian LCD 20x4, pengujian data pada PLX DAQ dan pengujian alat secara keseluruhan.

4.1 Pengujian *Fingerprint Scanner*

Fingerprint Scanner diuji dengan menggunakan program yang diupload ke dalam Arduino mega 2560. Ada beberapa program yang digunakan untuk menguji *fingerprint scanner*, diantaranya yaitu pendaftaran sidik jari (*enrollment*), pengujian kecocokan sidik jari. Sebelum menguji keberhasilan saat mendaftarkan sidik jari, perlu mengupload program library *enroll* terlebih dahulu. Program dapat dilihat pada Gambar 4.1

```
#include <Adafruit_Fingerprint.h>
SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial2);

uint8_t id;

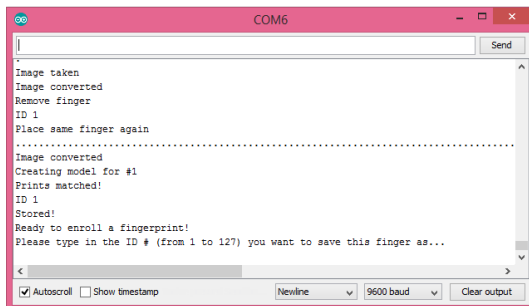
void setup()
{
  Serial.begin(9600);
  while (!Serial); // For Yun/Leo/Micro/Zero/...
  delay(100);
  Serial.println("\n\nAdafruit Fingerprint sensor enrollment");

  // set the data rate for the sensor serial port
  finger.begin(57600);

  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) { delay(1); }
  }
}
```

Gambar 4.1 Program *Enroll* untuk Daftar Sidik Jari

Setelah program terupload maka dapat mendaftarkan sidik jari dengan membuka serial monitor. Tampilan Serial monitor dapat dilihat pada Gambar 4.2



Gambar 4.2 Pengujian *Enrollment* pada *Fingerprint Scanner*

Pada pengujian Gambar 4.2 menunjukkan bahwa sidik jari telah berhasil mendapatkan *image* sidik jari. Maka sidik jari tersebut telah terdaftar dan tersimpan sesuai dengan ID yang dipilih.

Selanjutnya untuk mengecek bahwa sidik jari telah terdaftar sesuai dengan ID yang dipilih dan dapat mengetahui keakuratan *fingerprint*, dapat dicek dengan mengupload program *library fingerprint* pada arduino mega. Program dapat dilihat pada Gambar 4.3

```
#include <Adafruit_Fingerprint.h>

SoftwareSerial mySerial(16, 17);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial2);

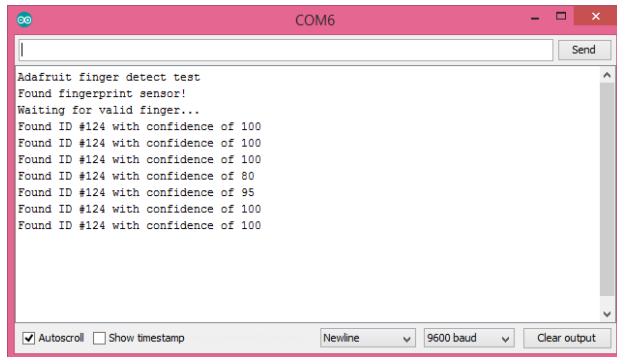
void setup()
{
  Serial.begin(9600);
  while (!Serial); // For Yun/Leo/Micro/Zero...
  delay(100);
  Serial.println("\n\nAdafruit finger detect test");

  // set the data rate for the sensor serial port
  finger.begin(57600);

  if (finger.verifyFasword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) { delay(1); }
  }
}
```

Gambar 4.3 Program untuk Mengecek Kecocokan Sidik Jari

Setelah program diupload, maka *login* terlebih dahulu coba menempelkan sidik jari pada sensor *fingerprint*, jika sidik jari terdeteksi maka dapat dilihat hasilnya pada serial monitor. Hasil pengujian kecocokan sidik jari yang ditampilkan pada serial monitor dapat dilihat pada Gambar 4.4



Gambar 4.4 Pengujian Kecocokan Sidik Jari

Hasil pengujian pada Gambar 4.4 membuktikan bahwa sidik jari telah terdaftar pada sensor *fingerprint*. Semakin tinggi nilai *confidence* pada serial monitor, maka semakin tinggi pula tingkat kecocokan sidik jari tersebut. Jadi untuk membuktikan bahwa sidik jari telah terdaftar yaitu dengan melihat serial monitor, jika saat sensor mendeteksi jari dan pada serial monitor muncul ID yang sesuai dengan ID yang telah didaftarkan, maka sidik jari telah terdeteksi dengan benar.

4.2 Pengujian LCD 20x4

Pengujian LCD ini bertujuan untuk memastikan apakah LCD dapat berfungsi dengan baik. Sehingga monitoring perhitungan jumlah barang pada konveyor dan hasil deteksi saat sidik jari berhasil terdeteksi oleh sensor *fingerprint* akan mendapatkan data yang baik. Untuk melakukan pengujian ini yaitu dengan membuat program terlebih dahulu. Untuk program yang digunakan untuk menampilkan tulisan menu sebelum *login* pada LCD 20x4 dapat dilihat pada Gambar 4.5.

```

//-----PRA LOGIN
if (proses == -2)
{
  //lcd.print("01234567890123456789");
  lcd.clear();
  if (status_hw_finger == 1)
  {
    lcd.setCursor(0, 0);
    lcd.print(" Place Your Finger ");
    lcd.setCursor(0, 1);
    lcd.print(" to Login ");
    lcd.setCursor(0, 3);
    lcd.print("Check-----Add User");
    proses = -1;
  }
  else
  {
    lcd.setCursor(0, 0);
    //lcd.print("01234567890123456789");
    lcd.print("Finger Sensor Error");
    lcd.setCursor(0, 3);
    lcd.print("Check---Manual ");
    proses = -1;
  }
}
}

```

Gambar 4.5 Program LCD untuk Tampil Menu Sebelum *Login*

Selanjutnya yaitu program yang telah dibuat akan di-*upload* ke Arduino. Setelah program ter-*upload*, maka sistem dapat dijalankan. Setelah ter-*upload* maka layar LCD akan menampilkan tampilan awal LCD yaitu menu sebelum *login*. Untuk itu dapat dilihat hasil pengujian pada Gambar 4.6



Gambar 4.6 Tampilan Awal LCD untuk *Login*

Hasil pengujian LCD yang terdapat pada Gambar 4.6 sudah sesuai dengan program yakni dapat menampilkan perintah untuk melakukan *login* terlebih dahulu. Karena untuk mengaktifkan mesin harus *login* dahulu. *Login* ini dengan menempelkan sidik jari pada sensor *fingerprint*. Ketika sensor *fingerprint* berhasil mendeteksi maka pada LCD akan muncul pemberitahuan bahwa *login* telah berhasil

seperti pada Gambar 4.7. Hasil pengujian tersebut sesuai dengan program yang kami buat.



Gambar 4.7 Tampilan LCD Ketika *Login* Berhasil

Ketika *login*, tetapi sidik jari belum terdaftar maka akan ada pemberitahuan yang ditampilkan pada LCD yaitu “*Login Failed*”. Untuk itu tampilan LCD dapat dilihat pada Gambar 4.8



Gambar 4.8 Tampilan LCD Ketika *Login* Gagal

Setelah proses *Login* Berhasil, kemudian barang diletakkan di ujung konveyor tepat didepan sensor dioda laser dan barang berjalan melewati sensor untuk mendeteksi barang berdasarkan ketinggian. Untuk mengecek jumlah barang yang terhitung yaitu lihat pada layar LCD. Sebelumnya, program harus diupload terlebih dahulu, program pada Gambar 4.9

```
//-----TAMPIL LCD PROSES
if (proses >= 0)
{
    itampil++;
    if (itampil > 100)
    {
        itampil = 0;
        lcd.setCursor(0, 0 );
        lcd.print("nQB="); lcd.print(nQrB); lcd.print(" ");
        lcd.setCursor(0, 1 );
        lcd.print("nQM="); lcd.print(nQrM); lcd.print(" ");
        lcd.setCursor(0, 2 );
        lcd.print("nQS="); lcd.print(nQrS); lcd.print(" ");
        lcd.setCursor(0, 3 );
        nQrT = nQrB + nQrM + nQrS;
        lcd.print("nQT="); lcd.print(nQrT); lcd.print(" ");

        lcd.setCursor(7, 0 );
        lcd.print("nSB="); lcd.print(nStB); lcd.print(" ");
        lcd.setCursor(7, 1 );
        lcd.print("nSM="); lcd.print(nStM); lcd.print(" ");
        lcd.setCursor(7, 2 );
        lcd.print("nSS="); lcd.print(nStS); lcd.print(" ");
        lcd.setCursor(7, 3 );
    }
}
```

Gambar 4.9 Program Tampilan LCD Jumlah Barang

Setelah program *terupload*, maka melakukan pengujian LCD, LCD dapat menampilkan data jumlah barang kecil, sedang, besar dan juga jumlah barang secara keseluruhan seperti pada Gambar 4.10 dan 4.11 Pada Gambar 4.10 LCD menampilkan hasil hitungan untuk barang kecil sebanyak satu kali, dan untuk total barangnya sebanyak 1 barang juga. Sedangkan pada Gambar 4.10 dapat diketahui bahwa LCD 20x4 menampilkan perhitungan dari ketiga macam barang, yaitu barang rendah, sedang dan tinggi serta total barang.



Gambar 4.10 Tampilan LCD saat Mendeteksi Barang Kecil



Gambar 4.11 Tampilan Ketika Semua Sensor Mendeteksi

Dari hasil pengujian untuk LCD 20x4 mendapatkan hasil yang sesuai. Dan ketika sensor mendeteksi sesuai dengan jenis barang, maka jumlah barang pada LCD langsung bertambah. Begitu juga untuk total barang secara keseluruhan, otomatis menjumlahkan dari jumlah masing-masing barang seperti pada Gambar 4.11

4.3 Pengujian Penambahan Sidik Jari Baru

Untuk dapat melakukan proses pengujian ini, *upload* terlebih dahulu program untuk mendaftarkan sidik jari baru pada arduino mega. Jadi, supaya untuk mendaftarkan sidik jari baru kita tidak perlu mengubah program lagi. Cukup memilih pilihan menu yang ada pada LCD untuk mendaftarkan sidik jari baru. Untuk itu kita atur programnya terlebih dahulu supaya penambahan sidik jari baru hanya dengan menekan menu pada LCD dan proses pendaftaran hanya dipantau di LCD hingga pendaftaran berhasil. Program dapat dilihat pada Gambar 4.12

```

//-----ADD Finger
else if (proses == -21)
{
  //lcd.print("01234567890123456789");
  lcd.setCursor(0, 0);
  lcd.print("Select NIP");
  lcd.setCursor(0, 1);
  lcd.print("NIP="); lcd.print(nip[index_nip] );
  lcd.setCursor(0, 3);
  lcd.print("      Back      Next");
  if (digitalRead(ptombol2) == LOW)
  {
    proses = -2;
  }
  if (digitalRead(ptombol1) == LOW)
  {
    index_nip++;
    if (index_nip > 10)
      index_nip = 1;
    Serial.println(index_nip);
    delay(300);
  }
  if (digitalRead(ptombol3) == LOW)
  {
    proses = -211;
    lcd.clear();
  }
}

```

Gambar 4.12 Program untuk Menambahkan Sidik Jari Baru

Setelah diupload maka proses penambahan sidik jari dapat dilakukan. Untuk menguji penambahan sidik jari baru yang belum terdaftar yaitu tanpa perlu mengubah program. Tetapi hanya dengan menekan tombol *push button* C untuk memilih menu “Add User” yang ditampilkan pada LCD. Ilustrasi dapat dilihat pada Gambar 4.13



Gambar 4.13 Ilustrasi Pemilihan Menu “Add User”

Setelah memilih menu *Add User* maka muncul tampilan pilihan NIP. Pilihan No NIP antara 1110001-1110010. Untuk memilih NIP yaitu dengan menekan tombol A seperti pada Gambar 4.13



Gambar 4.14 Ilustrasi Pemilihan NIP

Pada Gambar 4.14 hasil pengujian sudah sesuai dengan program yang kami buat sebelumnya. Bahwa ketika ditekan tombol A, maka No NIP akan terus bertambah dari 1110001-1110010. Setelah NIP dipilih maka tekan tombol C untuk memilih menu “Next” yang akan masuk ke proses selanjutnya yaitu mendaftarkan sidik jari baru. Sebagai contoh NIP yang saya pilih yaitu 1110004. Untuk itu, dapat dilihat pada Gambar 4.15



Gambar 4.15 Pemilihan menu “Next” dengan Tombol C

Setelah tombol C ditekan untuk memilih menu “Next” maka akan muncul tampilan perintah untuk mendaftarkan sidik jari Admin terlebih dahulu. Pada sistem ini kami *setting* bahwa untuk mendaftarkan sidik jari baru harus Admin dulu yang menscan sidik jari supaya tidak sembarang karyawan yang bisa mendaftarkan sidik jari baru. Admin bertugas jika ada penambahan karyawan baru maka Admin perlu untuk memilihkan NIP supaya tidak sama dengan NIP karyawan lain yang telah terdaftar. Untuk itu dapat dilihat pada Gambar 4.16



Gambar 4.16 Proses Admin *Scanning* Sidik Jari

Setelah admin *scan* sidik jari, maka akan masuk ke proses selanjutnya yaitu perintah untuk *scan* sidik jari baru yang akan didaftarkan. Sistem ini saya atur bahwa dalam satu NIP yaitu 2 jari. Jadi ketika ada salah satu jari yang luka atau tidak bisa digunakan maka masih mempunyai cadangan jari yang telah terdaftar juga. Untuk itu dapat dilihat pada Gambar 4.17 dan Gambar 4.18



Gambar 4.17 *Scanning* Sidik Jari Baru Pertama



Gambar 4.18 *Scanning* Sidik Jari Baru Kedua

Pada Gambar 4.17 terlihat bahwa tampilan layar LCD yang menampilkan perintah untuk *scanning* sidik jari baru yang pertama. Dan setelah sidik jari pertama berhasil hingga muncul perintah lagi untuk *scanning* sidik jari baru kedua seperti pada Gambar 4.18. Jika pendaftaran sidik jari baru selesai maka akan muncul pemberitahuan pada LCD yang membuktikan bahwa sidik jari dengan NIP 1110004 telah berhasil didaftarkan dan telah disimpan seperti pada Gambar 4.19



Gambar 4.19 Pemberitahuan Sidik Jari Berhasil Terdaftar

Berdasarkan hasil pengujian yang tertera pada Gambar 4.12 sampai 4.19 yaitu terbukti benar sesuai dengan program yang kami buat. Mulai dari pemilihan menu “Add User” , pemilihan NIP, pendaftaran sidik jari baru hingga sidik jari baru berhasil terdaftar sesuai dengan NIP yang dipilih. Pengujian ini juga termasuk kedalam pengujian tombol yang terdapat pada panel *box* yang digunakan untuk memilih menu untuk mendaftarkan sidik jari baru. Di mana fungsi Tombol A yaitu untuk memilih no NIP dan untuk mengecek kondisi sensor, Tombol B untuk kembali ke menu awal dan Tombol C untuk memilih tulisan “Next” yang digunakan untuk *scanning* sidik jari baru dan untuk memilih menu “Logout”. Setelah diuji tombol-tombol tersebut sudah berfungsi dengan baik dan sesuai dengan program yang kami buat.

4.3 Pengujian Berdasarkan Posisi Sidik Jari

Pengujian ini ditujukan untuk mengetahui seberapa jauh sistem ini dapat mengenali sidik jari dengan posisi yang berbeda beda. Begitu juga kami menguji waktu yang dibutuhkan untuk mendeteksi masing-masing jari. Hal ini untuk mengetahui seberapa cepat sistem ini bekerja. Pada pengujian ini, akan diuji dari beberapa jari yang berbeda-beda. Pengujian ini saya menguji 10 *User*. Dimana masing-masing *user*nya yaitu 2 jari yang telah terdaftar. Pengujian posisi sidik jari dapat dilihat pada Gambar 4.20, 4.21 dan 4.22.



Gambar 4.20 Posisi Sidik Jari saat Lurus



Gambar 4.21 Posisi Sidik Jari Saat Miring Kiri



Gambar 4.22 Posisi Sidik Jari saat Miring Kanan

Sedangkan hasil pengujian berdasarkan posisi sidik jari dapat dilihat pada Tabel 4.1

Tabel 4.1 Hasil Pengujian Berdasarkan Posisi Sidik Jari

NIP	Sidik Jari	Posisi Sidik Jari	Keterangan	Waktu Terdeteksi
Admin	Jempol Kanan	Lurus	Terdeteksi	1,14 detik
		Miring Kanan	Terdeteksi	1,10 detik
		Miring kiri	Terdeteksi	0,89 detik
	Telunjuk Kanan	Lurus	Terdeteksi	0,82 detik
		Miring Kanan	Terdeteksi	1,01 detik
		Miring Kiri	Terdeteksi	1,01 detik
1110001 (Karyawan A)	Jempol Kiri	Lurus	Terdeteksi	1,15 detik
		Miring Kanan	Terdeteksi	1,18 detik
		Miring Kiri	Terdeteksi	1,20 detik
	Telunjuk Kiri	Lurus	Terdeteksi	0,93 detik
		Miring Kanan	Terdeteksi	1,12 detik
		Miring Kiri	Terdeteksi	1,12 detik
1110002 (Karyawan B)	Jempol Kanan	Lurus	Terdeteksi	0,95 detik
		Miring Kanan	Terdeteksi	1,20 detik
		Miring kiri	Terdeteksi	1,12 detik
	Telunjuk Kanan	Lurus	Terdeteksi	0,89 detik
		Miring kanan	Terdeteksi	1,30 detik
		Miring Kiri	Terdeteksi	1,28 detik
1110003 (Karyawan C)	Jempol Kiri	Lurus	Terdeteksi	1,28 detik
		Miring kiri	Terdeteksi	1,36 detik
		Miring Kanan	Terdeteksi	1,32 detik
	Telunjuk Kiri	Lurus	Terdeteksi	1,32 detik
		Miring kiri	Terdeteksi	1,20 detik
		Miring kanan	Terdeteksi	1,31 detik

NIP	Sidik Jari	Posisi Sidik Jari	Keterangan	Waktu Terdeteksi
1110004 (Karyawan D)	Jari Tengah	Lurus	Terdeteksi	1,25detik
		Miring kiri	Terdeteksi	1,36 detik
		Miring Kanan	Terdeteksi	1,20 detik
	Jari Manis	Lurus	Terdeteksi	1,33 detik
		Miring kiri	Terdeteksi	1,32 detik
		Miring kanan	Terdeteksi	1,31 detik
1110005 (Karyawan E)	Jempol Kanan	Lurus	Terdeteksi	1,16 detik
		Miring kiri	Terdeteksi	1,30 detik
		Miring Kanan	Terdeteksi	1,22 detik
	Telunjuk Kanan	Lurus	Terdeteksi	1,25 detik
		Miring kiri	Terdeteksi	1,32 detik
		Miring kanan	Terdeteksi	1,31 detik
1110006 (Karyawan F)	Jempol Kiri	Lurus	Terdeteksi	1,18detik
		Miring kiri	Terdeteksi	1,30 detik
		Miring Kanan	Terdeteksi	1,23 detik
	Telunjuk Kiri	Lurus	Terdeteksi	1,25 detik
		Miring kiri	Terdeteksi	1,32 detik
		Miring kanan	Terdeteksi	1,31 detik
1110007 (Karyawan G)	Jempol Kanan	Lurus	Terdeteksi	1,22 detik
		Miring kiri	Terdeteksi	1,36 detik
		Miring Kanan	Terdeteksi	1,14 detik
	Telunjuk Ksnsn	Lurus	Terdeteksi	1,33 detik
		Miring kiri	Terdeteksi	1,32 detik
		Miring kanan	Terdeteksi	1,31 detik

NIP	Sidik Jari	Posisi Sidik Jari	Keterangan	Waktu Terdeteksi
1110008 (Karyawan H)	Jempol Kiri	Lurus	Terdeteksi	1,25detik
		Miring kiri	Terdeteksi	1,33 detik
		Miring Kanan	Terdeteksi	1,20 detik
	Telunjuk Kiri	Lurus	Terdeteksi	1,33 detik
		Miring kiri	Terdeteksi	1,21 detik
		Miring kanan	Terdeteksi	1,28 detik
1110009 (Karyawan I)	Jari Tengah	Lurus	Terdeteksi	1,25detik
		Miring kiri	Terdeteksi	1,20 detik
		Miring Kanan	Terdeteksi	1,36 detik
	Jari Manis	Lurus	Terdeteksi	1,33 detik
		Miring kiri	Terdeteksi	1,32 detik
		Miring kanan	Terdeteksi	1,31 detik

Berdasarkan hasil pengujian pada Tabel 4.1 telah diuji 16 *user* dengan sidik jari orang yang berbeda-beda. Untuk 1 *User*nya ada 2 sidik jari. Supaya jika ada jari yang terluka tetap bisa *login* maupun *logout*. Jadi dapat disimpulkan bahwa untuk semua jari dapat dideteksi oleh sensor *fingerprnt* walaupun posisi sidik jari sedikit miring ke kanan maupun ke kiri. Asalkan posisi jari tersebut masih pas pada sensor *fingerprnt* tersebut. Sedangkan untuk waktu yang dibutuhkan sensor *fingerprnt* untuk mendeteksi sidik jari ini cukup cepat karena hanya membutuhkan waktu sekitar 0.82 – 1.36 detik. Kecepatan waktu membaca tersebut tergantung dari posisi sidik jari yang kurang pas maupun kondisi jari yang kurang bersih. Hal itu bisa menyebabkan lama atau cepatnya sensor *fingerprnt* mendeteksi sidik jari. Setelah itu kami juga menguji saat kondisi tangan berbeda-beda yaitu bersih dan berminyak. Bersih itu berarti tangan keadaan kering. Hasil pengujian dapat dilihat pada Tabel 4.2

Tabel 4.2 Hasil Pengujian Berdasarkan Kondisi Sidik Jari

NIP	Sidik Jari	Kondisi Sidik Jari	Keterangan	Waktu Terdeteksi
Admin	Jempol Kanan	Bersih	Terdeteksi	0,63 detik
		Berminyak	Terdeteksi	0,83 detik
	Telunjuk Kanan	Bersih	Terdeteksi	0,66 detik
		Berminyak	Tidak Terdeteksi	-
1110001	Jempol Kiri	Bersih	Terdeteksi	0,63 detik
		Berminyak	Tidak Terdeteksi	-
	Telunjuk Kiri	Bersih	Terdeteksi	0,66 detik
		Berminyak	Terdeteksi	0,83 detik
1110002	Jempol Kanan	Bersih	Terdeteksi	0,88 detik
		Berminyak	Terdeteksi	0,82 detik
	Telunjuk Kanan	Bersih	Terdeteksi	0,63 detik
		Berminyak	Terdeteksi	0,97 detik
1110003	Jempol Kiri	Bersih	Terdeteksi	0,72 detik
		Berminyak	Terdeteksi	0,82 detik
	Telunjuk Kiri	Bersih	Terdeteksi	0,75 detik
		Berminyak	Tidak Terdeteksi	-
1110004	Jari Tengah	Bersih	Terdeteksi	0,72 detik
		Berminyak	Terdeteksi	0,99 detik
	Jari Manis	Bersih	Terdeteksi	0,68 detik
		Berminyak	Terdeteksi	0,95 detik
1110005	Jempol Kiri	Bersih	Terdeteksi	0,63 detik
		Berminyak	Tidak Terdeteksi	0,83 detik
	Telunjuk Kiri	Bersih	Terdeteksi	0,66 detik
		Berminyak	Terdeteksi	-

NIP	Sidik Jari	Kondisi Sidik Jari	Keterangan	Waktu Terdeteksi
1110006	Jempol Kanan	Bersih	Terdeteksi	0,63 detik
		Berminyak	Terdeteksi	-
	Telunjuk Kanan	Bersih	Terdeteksi	0,66 detik
		Berminyak	Tidak Terdeteksi	0,85 detik
1110007	Jempol Kiri	Bersih	Terdeteksi	0,63 detik
		Berminyak	Tidak Terdeteksi	0,78 detik
	Telunjuk Kiri	Bersih	Terdeteksi	0,63 detik
		Berminyak	Terdeteksi	0,92 detik
1110008	Jempol Kanan	Bersih	Terdeteksi	0,88 detik
		Berminyak	Terdeteksi	0,82 detik
	Telunjuk Kanan	Bersih	Terdeteksi	0,63 detik
		Berminyak	Terdeteksi	0,97 detik
1110009	Jempol Kiri	Bersih	Terdeteksi	0,72 detik
		Berminyak	Terdeteksi	-
	Telunjuk Kiri	Bersih	Terdeteksi	0,75 detik
		Berminyak	Tidak Terdeteksi	0,95 detik

Berdasarkan hasil pengujian pada Tabel 4.2 dapat disimpulkan bahwa untuk semua jari dapat dideteksi oleh sensor *fingerprint* walaupun kondisi sidik jari dalam keadaan bersih maupun berminyak. Asalkan posisi jari tersebut masih pas pada sensor *fingerprint* tersebut. Kondisi tangan bersih itu berarti tangan dalam keadaan kering. Sedangkan berminyak saya menguji dengan mengoleskan sedikit minyak zaitun pada sidik jari. Berdasarkan hasil pengujian pada Tabel 4.2 yaitu dapat disimpulkan bahwa saat kondisi tangan bersih maka sensor *fingerprint* selalu mendeteksi dengan baik. Sedangkan saat kondisi tangan berminyak sensor *fingerprint* ini tetap bisa mendeteksi sidik jari, tetapi dari 20 kali percobaan ada 6 kali tidak terdeteksi. Hal itu mungkin karena minyak pada jari terlalu banyak sehingga tidak bisa untuk mendeteksi. Jika dilihat kecepatan saat sensor *fingerprint* mendeteksi sidik jari dalam kondisi bersih dan berminyak ini. Waktu

mendeteksinya lebih cepat ketika tangan bersih yaitu 0,63-0,88 detik. Sedangkan waktu mendeteksi sidik jari dalam kondisi berminyak yaitu 0,82-0,99 detik.

Untuk itu dapat dihitung persentase *error* yang didapat dari data pengujian kondisi sidik jari, dengan menggunakan rumus sebagai berikut :

$$\text{Persentase Error} = \frac{(\text{Nilai Sebenarnya}) - (\text{Nilai terbaca})}{(\text{Nilai Sebenarnya})} \times 100\% \dots\dots (1)$$

$$\begin{aligned} \text{Persentase Error} &= \frac{(80-74)}{80} \times 100\% \\ &= \frac{6}{80} \times 100\% \\ &= 7,5 \% \end{aligned}$$

Hasil dari presentase *error* di atas menunjukkan bahwa *fingerprint scanner* terdapat presentase *error* sebesar 7,5 %. Sehingga keakuratan pembacaan sensor *fingerprint* sebesar 92,5 %.

4.3 Pengujian Ketepatan Deteksi Sensor Dioda Laser

Pengujian sensor dioda laser ini digunakan untuk mengetahui seberapa jauh sensor dioda laser dapat mendeteksi barang berdasarkan ketinggian. Pengujian ini dapat juga untuk menguji ketepatan setiap sensor dioda laser saat mendeteksi barang. Sebelum itu kami melakukan pengujian hingga jarak dari sensor ke barang maksimal 18 cm dan sensor dioda laser tetap bisa mendeteksi. Jadi ketika barang melewati konveyor maka sensor didoa laser akan terhalang oleh barang tersebut. Jika barang kecil yang sedang melewati sensor dioda laser maka sensor Ss aktif. Ketika barang sedang yang melewati sensor dioda laser maka sensor Ss dan Sm aktif. Dan saat barang besar melewati sensor dioda laser maka sensor Ss,Sm,dan Sb aktif. Dimana Ss yaitu sensor rendah, Sm yaitu sensor sedang, dan Sb yaitu sensor tinggi. Untuk bisa melakukan proses tersebut harus mengupload program terlebih dahulu pada Arduino Mega. Program dapat dilihat pada Gambar 4.23

```

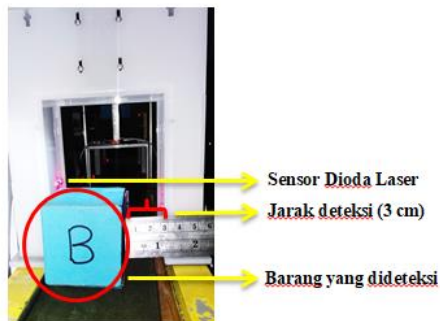
if (dsensor_big == sensor_terhalang && dsensor_medium == sensor_terhalang && dsensor_small == sensor_terha
{
    benda = big;
    nStB++;
}
else if (dsensor_medium == sensor_terhalang && dsensor_small == sensor_terhalang )
{
    benda = medium;
    nStM++;
}
else if (dsensor_small == sensor_terhalang )
{
    nStS++;
    benda = small;
}
}

if ( benda == big)
{
    lcd.setCursor(14, 2 );
    lcd.print(last_Qrfull);
    lcd.setCursor(14, 3 );
    lcd.print("BIG  ");
}

```

Gambar 4.23 Program Deteksi Barang Berdasarkan Ketinggian

Setelah program *terupload* pada arduino mega, maka dapat melakukan pengujian jarak deteksi sensor dioda laser. Jarak tersebut diukur dari sensor dioda laser ke barang. Hal ini untuk mengetahui pada jarak berapa saja sensor dapat aktif. Untuk melihat pengukuran jarak dari sensor dioda laser ke barang yaitu dapat dilihat pada Gambar 4.24



Gambar 4.24 Salah Satu Pengujian Jarak Deteksi Sensor

Dari pengujian jarak deteksi dari sensor ke barang dapat diketahui hasilnya pada Tabel 4.3, Tabel 4.4 dan Tabel 4.5.

Tabel 4.3 Hasil Pengujian Ketepatan Deteksi Sensor Rendah

No	Barang yang dideteksi	Jarak deteksi (cm)	Barang Hasil Deteksi	Keterangan
1.	Kecil	3 cm	Kecil	Tepat
2.	Kecil	2,5 cm	Kecil	Tepat
3	Kecil	2 cm	Kecil	Tepat
4.	Kecil	1,5 cm	Kecil	Tepat
5.	Kecil	1 cm	Kecil	Tepat
6.	Kecil	0,5 cm	Kecil	Tepat

Tabel 4.4 Hasil Pengujian Ketepatan Deteksi Sensor Sedang

No	Barang yang dideteksi	Jarak deteksi (cm)	Barang Hasil Deteksi	Keterangan
1.	Sedang	3 cm	Sedang	Tepat
2.	Sedang	2,5 cm	Sedang	Tepat
3	Sedang	2 cm	Sedang	Tepat
4.	Sedang	1,5 cm	Sedang	Tepat
5.	Sedang	1 cm	Sedang	Tepat
6.	Sedang	0,5 cm	Sedang	Tepat

Tabel 4.5 Hasil Pengujian Ketepatan Deteksi Sensor Tinggi

No	Barang yang dideteksi	Jarak deteksi (cm)	Barang Hasil Deteksi	Keterangan
1.	Besar	3 cm	Besar	Tepat
2.	Besar	2,5 cm	Besar	Tepat
3	Besar	2 cm	Besar	Tepat
4.	Besar	1,5 cm	Besar	Tepat
5.	Besar	1 cm	Besar	Tepat
6.	Besar	0,5 cm	Besar	Tepat

Dari hasil pengujian pada Tabel 4.3, Tabel 4.4, dan Tabel 4.5 didapatkan bahwa sensor dioda laser telah mendeteksi barang sesuai dengan ketinggian dengan tepat. Pada hasil pengujian ini, terbukti bahwa sensor dioda laser ini bisa mendeteksi barang dari jarak berapapun, karena setelah saya coba hingga jarak 18 cm sensor dioda laser tetap bisa mendeteksi. Jadi dapat disimpulkan bahwa jarak tidak mempengaruhi sensor dioda laser untuk mendeteksi barang. Kebetulan saja jarak dari barang ke sensor dioda laser yang terpasang dari konveyor maksimal hanya 3 cm. Jadi, dari jarak 0,5 – 3 cm sensor dapat mendeteksi barang dengan sesuai. Salah satu contoh barang besar telah melewati sensor dioda laser dan sensor tersebut terhalang oleh benda, maka sensor 1,2,dan 3 akan aktif yang hasil deteksinya juga tergolong besar. Hal itu menunjukkan bahwa hasil deteksinya tepat. Untuk melihat ketepatan sensor saat mendeteksi barang dapat dilihat hasil tampilan pada LCD. Apabila barang yang dideteksi merupakan barang kecil maka tampilan di LCD untuk barang kecil (nSS) dan total keseluruhan barang (nST) juga akan berubah. Untuk melihat hasil sensor saat mendeteksi barang dapat dilihat pada Gambar 4.25



Gambar 4.25 Tampilan LCD Saat Sensor Mendeteksi Barang

4.4 Pengujian Waktu Sensor Dioda Laser Mendeteksi Barang

Pengujian waktu sensor dioda laser mendeteksi barang ini bertujuan untuk mengetahui waktu kemampuan sensor dalam mendeteksi. Langkah pertama yang dilakukan dalam pengujian ini yaitu menyiapkan komponen atau peralatan yang dibutuhkan, misalnya 3 buah sensor dioda laser, Arduino, *stopwatch* dan penggaris. Kemudian membuat program, program tersebut di-*upload* ke Arduino Mega. Setelah program di-*upload* , maka pengujian dapat dilakukan. Cara kerjanya yaitu apabila sensor mendeteksi barang rendah maka hanya sensor Ss saja yang aktif, apabila sensor mendeteksi barang sedang maka hanya sensor Ss dan Sm saja yang aktif, dan apabila sensor mendeteksi barang tinggi maka sensor Ss, Sm dan Sb aktif bersamaan.

Selanjutnya yaitu melakukan pengujian dengan cara menjalankan barang dari ujung konveyor hingga barang melewati sensor dioda laser yang akan mendeteksi barang tersebut. Letak barang dengan sensor dioda laser diukur menggunakan penggaris. Ketika barang diletakkan pada ujung konveyor, *stopwatch* mulai dinyalakan sampai barang terdeteksi oleh sensor. Dan untuk ilustrasi pengujian ini dapat dilihat pada Gambar 4.26



Gambar 4.26 Salah Satu Pengujian Waktu Deteksi Sensor

Dari Gambar 4.26 dapat dilihat bahwa dalam pengujian waktu deteksi sensor dioda laser untuk mendeteksi barang yaitu dengan cara meletakkan salah satu jenis barang yang akan dideteksi. Barang diletakkan dengan jarak tertentu. Jarak deteksi dapat diukur menggunakan penggaris. Jarak deteksi yang dimaksud merupakan jarak dari sensor hingga barang yang dideteksi. Dan untuk hitungan *stopwatch* dimulai saat barang diletakkan pada ujung konveyor dan diakhiri saat sensor mendeteksi. Untuk melihat hasil pengujian kecepatan sensor saat mendeteksi barang yang diukur dengan menggunakan stopwatch dapat dilihat pada Tabel 4.6

Tabel 4.6 Hasil Pengujian Waktu Deteksi Sensor

No	Jenis Barang yang Dideteksi	Jarak Deteksi (cm)	Waktu Mendeteksi (detik)
1.	Kecil	3 cm	5,23
2.	Kecil	2,5 cm	5,32
3.	Kecil	2 cm	5,43
4.	Sedang	3 cm	5,35
5.	Sedang	2,5 cm	5,29

No	Jenis Barang yang Dideteksi	Jarak Deteksi (cm)	Waktu Mendeteksi (detik)
6.	Sedang	2 cm	5,39
7.	Besar	3 cm	5,22
8.	Besar	2,5 cm	5,16
9.	Besar	2 cm	5,30

Berdasarkan data dari hasil pengujian pada Tabel 4.6 didapatkan hasil kecepatan sensor dioda laser saat mendeteksi barang. Pengukuran kecepatan diukur dari saat barang diletakkan dari ujung konveyor hingga sensor dioda laser yang disusun secara vertikal mendeteksi barang. Sensor dioda laser ketika mendeteksi barang kecil membutuhkan waktu 5,32 – 5,43 detik, tergantung pada jarak deteksi. Dan ketika sensor inframerah mendeteksi barang sedang membutuhkan waktu sekitar 5,29 – 5,39 detik, tergantung pada jarak deteksi barang. Dan untuk barang besar membutuhkan waktu sekitar 5,16-5,30 detik tergantung pada jarak deteksi barang.

4.5 Pengujian Sistem Keseluruhan

Setelah dilakukan pengujian pada setiap bagian sistem, maka perlu dilakukan pengujian sistem keseluruhan. Pengujian sistem secara keseluruhan ini bertujuan untuk mengetahui berhasil tidaknya dari alat yang telah dibuat.

Langkah pertama untuk melakukan pengujian ini yaitu dengan membuat program Arduino untuk pengujian. Program Arduino yang digunakan dapat dilihat pada Gambar 4.1, 4.3, 4.5, 4.9, 4.12, 4.23, 4.36, 4.42, 4.47 dan 4.50. Selanjutnya yaitu merangkai komponen yang dibutuhkan mulai dari sensor *fingerprint*, sensor dioda laser, Arduino, LCD 20x4, *relay*, *Button* dan Motor DC. Untuk rangkaian sensor *Fingerprint* ke Arduino dapat dilihat pada Gambar 3.4. Untuk rangkaian sensor dioda laser ke arduino dapat dilihat pada Gambar 3.5. Untuk LCD 20x4 ke Arduino dapat dilihat pada Gambar 3.6. Untuk rangkaian Arduino ke *relay* dapat dilihat Gambar 3.7, sedangkan untuk rangkaian *relay* ke motor dapat dilihat pada Gambar 3.8. Untuk rangkaian *relay* ke *power supply* dapat dilihat pada Gambar 3.9. Untuk rangkaian *button* dengan arduino. Dapat dilihat pada Gambar 3.10. Setelah semua rangkaian sudah tersambung dengan benar maka akan dilakukan proses pengujian keseluruhan. Pengujian keseluruhan ini mulai dari *login* dengan sensor *fingerprint* untuk mengaktifkan mesin.

Setelah mesin aktif, memulai untuk pengujian dari barang yang berada pada ujung konveyor pertama untuk dideteksi oleh sensor dioda laser. Ketika barang telah dideteksi oleh sensor dioda laser maka konveyor akan bergerak dan barang berjalan hingga melewati sensor dioda laser yang disusun secara vertikal untuk mendeteksi ketinggian barang. Setelah semua proses selesai maka akan muncul hasil perhitungan barang. Hal pertama yang dilakukan adalah menyambungkan konveyor dengan sumber tegangan. Kemudian tombol *power* yang ada pada konveyor dinyalakan. Kondisi awal dari alat ini yaitu LCD dan sensor-sensor ini telah aktif. Namun konveyor tidak dapat berjalan jika belum *login*. Saat tombol *power* telah dinyalakan maka Layar LCD menampilkan karakter sesuai dengan program yang telah dibuat seperti pada Gambar 4.6 dan dalam beberapa detik tampilan awal konveyor berubah seperti pada Gambar 4.27



Gambar 4.27 Tampilan Awal Layar LCD 20x4



Gambar 4.28 Tampilan Awal Perintah *Login* di LCD

Tampilan awal layar LCD 20x4 memunculkan karakter berupa tulisan “MONITORING JUMLAH BARANG” sekitar 5 detik kemudian berganti dengan karakter tulisan “Place Your Finger to Login” yang berarti perintah menempelkan sidik jari pada *fingerprint* untuk melakukan *login* terlebih dahulu. Karena jika belum *login* maka mesin konveyor tidak akan berjalan. Setelah menempelkan jari pada

sensor *fingerprint* seperti pada Gambar 4.19, maka ketika *login* berhasil terdapat pemberitahuan yang ditampilkan pada LCD dengan karakter tulisan seperti pada Gambar 4.29. *Login* akan berhasil, ketika sidik jari sudah terdaftar pada sensor *fingerprint*.



Gambar 4.29 Tampilan LCD Ketika *Login* Berhasil

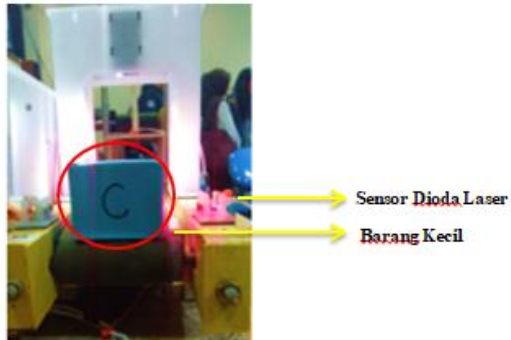
Setelah *login* berhasil, maka konveyor akan aktif (berjalan) ketika sensor telah mendeteksi barang yang ada di ujung konveyor 1. Pengujian yang pertama yaitu untuk pengelompokkan barang kecil. Untuk itu program dari pengelompokkan barang kecil dapat dilihat pada Gambar 4.30

```
//-----SMALL
else if (benda == small && proses == 3)//proses setelah sortir
{
    motor(motor_1_k_depan, nyala);
    motor(motor_2_k_lift, nyala);
    dsensor_2 = baca_sensor(psensor_2);
    if (dsensor_2 == sensor_terhalang)
    {
        motor(motor_2_k_lift, mati);
        motor(motor_1_k_depan, mati);
        delay(1000);
        proses = 311;

        timeup = millis();
        Serial.print("START :");
        Serial.print(timeup);
        Serial.print(" ms \n");
    }
}
else if (proses == 311)//lift mulai naik
{
    lift_naik();
    if (dlimitlift_atas == 0)
    {
        unsigned long time_done = millis();
        Serial.print("STOP :");
        Serial.print(time_done);
    }
}
```

Gambar 4.30 Program Pengelompokkan Barang Kecil

Selanjutnya yaitu proses pengelompokkan barang kecil. Pertama yaitu barang kecil diletakkan pada ujung konveyor 1 untuk dideteksi oleh sensor dioda laser dan konveyor berjalan membawa barang tersebut. Untuk itu dapat dilihat seperti pada Gambar 4.31



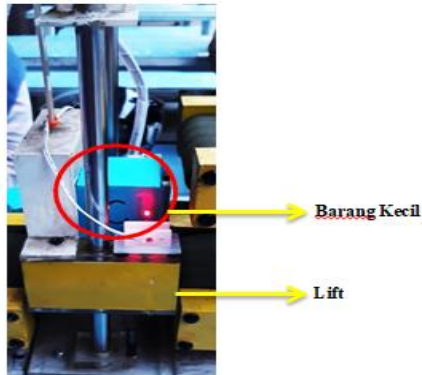
Gambar 4.31 Barang Kecil saat Terdeteksi Oleh Sensor 1

Setelah konveyor aktif, maka barang kecil berjalan melewati sensor dioda laser yang disusun secara vertikal. Disitu sensor dioda laser akan mendeteksi barang berdasarkan ketinggian. Barang kecil akan terdeteksi oleh sensor dioda laser 1 (paling bawah). Ilustrasi dapat dilihat pada Gambar 4.32 bahwa ketika sensor 1 itu terhalang maka sensor 1 tersebut aktif dan mendeteksi barang kecil untuk mencocokkan hasilnya dapat dicek pada layar LCD.



Gambar 4.32 Barang Kecil Terdeteksi Sensor Ketinggian

Setelah itu maka barang akan bergerak menuju konveyor 2. Setelah sampai di konveyor 2 maka motor 2 mati dan motor 3 untuk menggerakkan *lift* aktif. Sehingga untuk barang kecil didistribusikan dengan menaiki *lift*.



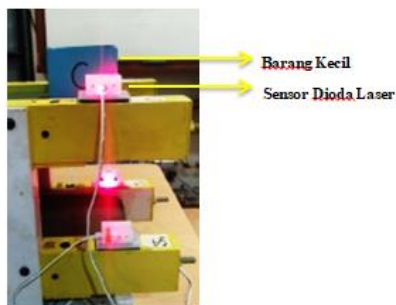
Gambar 4.33 Barang Kecil Melewati Konveyor 2 dan Naik Ke *Lift*

Pada Gambar 4.33 menunjukan bahwa barang kecil diangkut naik keatas menggunakan *lift*. Untuk menaiki *lift* tersebut barang yang diangkut untuk naik keatas menggunakan *lift* yaitu idealnya dengan berat 14 gram (*box* barang dari kertas duplex yang telah saya buat). Karena semakin berat barang yang diangkut untuk naik *lift* maka akan mempengaruhi kecepatan *lift* untuk naik. Jadi waktunya semakin lama untuk membawa barang tersebut naik keatas. Untuk itu dapat dilihat pada Tabel 4.7 yaitu pengujian kecepatan saat naik keatas *lift* berdasarkan berat barang.

Tabel 4.7 Pengujian Kecepatan Naik *Lift* Berdasarkan Berat Barang

Jenis Barang	Berat Barang	Waktu Naik Keatas <i>Lift</i>
Box Barang 1	14 gram	15,26 detik
Box Barang 2	58 gram	15,47 detik
Box Barang 3	116 gram	15,82 detik
Box Barang 4	313 gram	18,42 detik
Box Barang 5	370 gram	18,76 detik

Setelah sampai diatas maka *lift* berhenti dan konveyor 2 bergerak membawa barang menuju konveyor 4. Ketika barang terdeteksi oleh sensor dioda laser di konveyor 4 maka konveyor 2 mati dan *lift* akan turun. Sehingga barang jatuh ke bawah. Jadi, Barang kecil dikelompokkan ke kategori naik keatas dengan *lift*. Untuk itu dapat dilihat pada Gambar 4.34



Gambar 4.34 Barang Melewati Konveyor 4

Untuk itu tampilan LCD saat barang kecil berhasil dikelompokkan yaitu seperti Gambar 4.35



Gambar 4.35 Tampilan LCD saat Mendeteksi Barang Kecil

Pada Gambar 4.35 tampilan LCD ketika barang kecil berhasil dideteksi maka nSS yang berarti jumlah barang kecil bertambah 1 dan nST yang berarti total barang keseluruhan pun ikut bertambah 1.

Untuk melihat hasil pengujian pengelompokkan barang kecil yang dideteksi oleh sensor dioda laser dapat dilihat pada Tabel 4.8.

Tabel 4.8 Hasil Pengujian Pengelompokkan Barang Kecil

Barang yang Diuji	Barang Hasil Pengujian	Keterangan
Barang Kecil	Barang Kecil	Dikelompokkan dengan tepat
Barang Kecil	Barang Kecil	Dikelompokkan dengan tepat
Barang Kecil	Barang Kecil	Dikelompokkan dengan tepat
Barang Kecil	Barang Kecil	Dikelompokkan dengan tepat
Barang Kecil	Barang Kecil	Dikelompokkan dengan tepat
Barang Kecil	Barang Kecil	Dikelompokkan dengan tepat
Barang Kecil	Barang Kecil	Dikelompokkan dengan tepat

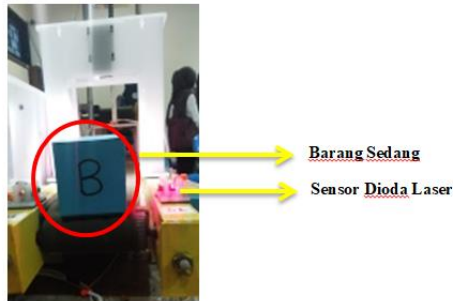
Selanjutnya yaitu pengujian untuk barang sedang. Program untuk pengelompokkan barang sedang yaitu dapat dilihat pada Gambar 4.36.

```
.  
//-----MEDIUM  
else if (benda == medium && proses == 3)  
{  
    motor(motor_1_k_depan, nyala);  
    motor(motor_2_k_lift, nyala);  
    dsensor_2 = baca_sensor(psensor_2);  
    if (dsensor_2 == sensor_terhalang)  
    {  
        motor(motor_5_k_bawah, nyala);  
        motor(motor_1_k_depan, mati);  
        delay(1000);  
        motor(motor_2_k_lift, mati);  
        proses = 321;  
    }  
}  
else if (proses == 321)//proses barang medium di konveyor 3  
{  
    delay(1700);  
    motor(motor_5_k_bawah, mati);  
    proses = 322;  
}  
else if (proses == 322)//proses barang medium didorong  
{  
    motor_push_dorong();  
    if (dlimitdorong_belakang == 1)
```

Gambar 4.36 Program Mengelompokkan Barang Sengah

Selanjutnya yaitu proses pengelompokkan barang sedang. Pertama yaitu barang sedang diletakkan pada ujung konveyor 1 untuk

dideteksi oleh sensor dioda laser dan konveyor berjalan membawa barang sedang tersebut. Untuk itu dapat dilihat seperti pada Gambar 4.37.



Gambar 4.37 Barang Sedang Terdeteksi oleh Sensor di Konveyor 1

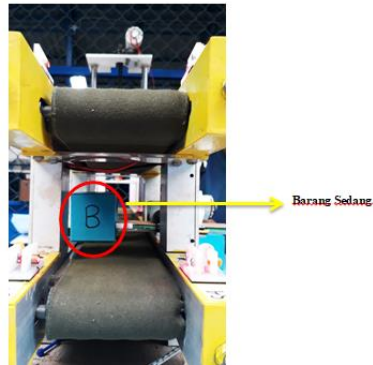
Setelah konveyor aktif, maka barang sedang berjalan melewati sensor dioda laser yang disusun secara vertikal. Disitu sensor dioda laser akan mendeteksi barang berdasarkan ketinggian. Barang sedang akan terdeteksi oleh sensor dioda laser 1 (paling bawah) dan dioda laser 2 (tengah). Untuk itu dapat dilihat pada Gambar 4.38 bahwa ketika sensor 1 dan 2 itu terhalang maka sensor 1 dan 2 tersebut aktif dan mendeteksi barang kecil untuk mencocokkan hasilnya dapat dicek pada layar LCD.



Gambar 4.38 Barang Sedang Terdeteksi oleh Sensor Ketinggian

Setelah itu maka barang akan bergerak menuju konveyor 2. Saat barang sedang terdeteksi oleh sensor 2 pada konveyor 2 maka

konveyor 3 aktif dan barang menuju konveyor 3. Selang 1,7 detik barang berhenti dan motor 6 aktif untuk menggerakkan *pusher*. *Pusher* ini untuk mendorong barang sedang menuju keluar. Ilustrasi dapat dilihat pada Gambar 4.39



Gambar 4.39 Barang Sedang Akan Didorong oleh *Pusher*

Pada Gambar 4.39 menunjukkan bahwa barang sedang didorong oleh *pusher* keluar. Jadi, barang sedang dikelompokkan ke kategori terdorong keluar oleh *pusher* dan berada di turunan warna bening seperti pada Gambar 4.40



Gambar 4.40 Barang Sedang Jatuh didorong oleh *Pusher*

Untuk itu tampilan LCD ketika mendeteksi barang sedang yaitu dapat dilihat pada Gambar 4.41



Gambar 4.41 Tampilan saat Barang Sedang Terdeteksi Sensor

Pada Gambar 4.41 tampilan LCD ketika barang sedang berhasil dideteksi maka nSM yang berarti jumlah barang sedang bertambah 1 dan nST yang berarti total barang keseluruhan pun ikut bertambah 1. Untuk itu hasil Pengujian dapat dilihat pada Tabel 4.9

Tabel 4.9 Hasil Pengujian Pengelompokkan Barang Sedang

Barang yang Diuji	Barang Hasil Pengujian	Keterangan
Barang Sedang	Barang Sedang	Dikelompokkan dengan tepat
Barang Sedang	Barang Sedang	Dikelompokkan dengan tepat
Barang Sedang	Barang Sedang	Dikelompokkan dengan tepat
Barang Sedang	Barang Sedang	Dikelompokkan dengan tepat
Barang Sedang	Barang Sedang	Dikelompokkan dengan tepat
Barang Sedang	Barang Sedang	Dikelompokkan dengan tepat
Barang Sedang	Barang Sedang	Dikelompokkan dengan tepat

Pengujian yang selanjutnya yaitu untuk barang besar. Untuk program mengelompokkan barang besar dapat dilihat pada Gambar 4.42.

```
//-----BIG
else if (benda == big && proses == 3)
{
    //motor(motor_1_k_depan, nyala);
    motor(motor_2_k_lift, nyala);
    dsensor_2 = baca_sensor(psensor_2);
    if (dsensor_2 == sensor_terhalang)
    {
        motor(motor_5_k_bawah, nyala);
        motor(motor_1_k_depan, mati);
        delay(1000); //untuk memastikan barang sudah berpindah ke konveyor 3
        motor(motor_2_k_lift, mati);
        proses = 331;
    }
}
else if (proses == 331)
{
    dsensor_4 = baca_sensor(psensor_4);
    if (dsensor_4 == sensor_terhalang)
    {
        delay(1000);
        motor(motor_5_k_bawah, mati);
        proses = 0; inputString = "";
    }
}
```

Gambar 4.42 Program Mengelompokkan Barang Besar

Selanjutnya yaitu, proses pengelompokkan barang besar. Pertama diletakkan pada ujung konveyor 1 untuk dideteksi oleh sensor dioda laser dan konveyor berjalan membawa barang tersebut. Untuk ilustrasinya dapat dilihat seperti pada Gambar 4.43



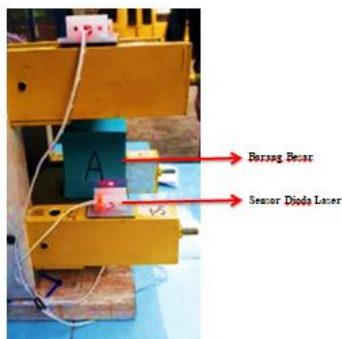
Gambar 4.43 Barang Besar Terdeteksi Sensor di Konveyor 1

Setelah konveyor aktif, maka barang besar berjalan melewati sensor dioda laser yang disusun secara vertikal. Disitu sensor dioda laser akan mendeteksi barang berdasarkan ketinggian. Barang besar akan terdeteksi oleh sensor dioda laser 1 (paling bawah), dioda laser 2 (tengah) dan sensor dioda laser 3 (paling atas). Untuk itu dapat dilihat pada Gambar 4.44 bahwa ketika sensor 1,2, dan 3 itu terhalang maka sensor 1,2, dan 3 tersebut aktif dan mendeteksi barang besar untuk mencocokkan hasilnya dapat dicek pada layar LCD



Gambar 4.44 Barang Besar Terdeteksi Sensor Dioda Laser

Setelah itu maka barang akan bergerak menuju konveyor 2. Saat barang sedang terdeteksi oleh sensor 2 pada konveyor 2 maka konveyor 3 aktif dan barang berjalan menuju konveyor 3.



Gambar 4.45 Barang Besar di Konveyor 3

Pada Gambar 4.45 terlihat bahwa ketika barang besar terdeteksi oleh sensor dioda laser di ujung konveyor 3 maka konveyor 3 mati dan barang terjatuh kebawah. Jadi, barang besar dikelompokkan lurus. Untuk ilustrasi saat barang besar terjatuh dari konveyor 3 dapat dilihat pada Gambar 4.46



Gambar 4.46 Barang Besar Terjatuh Kebawah

Setelah barang besar selesai dikelompokkan, maka jumlah barang dapat dilihat pada layar LCD seperti pada Gambar 4.47



Gambar 4.47 Tampilan LCD saat Barang Terdeteksi Sensor

Pada Gambar 4.47 tampilan LCD ketika barang besar berhasil dideteksi maka nSB yang berarti jumlah barang besar bertambah 1 dan nST yang berarti total barang keseluruhan pun ikut bertambah 1. Maka hasil pengujian pengelompokkan barang besar dapat dilihat pada Tabel 4.10

Tabel 4.10 Hasil Pengujian Pengelompokkan Barang Besar

Barang yang Diuji	Barang Hasil Pengujian	Keterangan
Barang Besar	Barang Besar	Dikelompokkan dengan tepat
Barang Besar	Barang Besar	Dikelompokkan dengan tepat

Barang yang Diuji	Barang Hasil Pengujian	Keterangan
Barang Besar	Barang Besar	Dikelompokkan dengan tepat
Barang Besar	Barang Besar	Dikelompokkan dengan tepat
Barang Besar	Barang Besar	Dikelompokkan dengan tepat
Barang Besar	Barang Besar	Dikelompokkan dengan tepat
Barang Besar	Barang Besar	Dikelompokkan dengan tepat

Berdasarkan hasil pengujian dari Tabel 4.8, Tabel 4.9 dan tabel 4.10 terlihat bahwa hasil pengujian pengelompokkan barang telah berhasil dan tepat. Dikarenakan ketika barang kecil sudah terdeteksi oleh sensor dioda laser yang disusun secara vertikal. Maka barang dikelompokkan naik keatas dengan *lift*. Ketika barang sedang terdeteksi oleh sensor dioda laser maka barang dikelompokkan keluar yang didorong oleh *pusher*. Ketika barang besar terdeteksi oleh sensor dioda laser maka barang dikelompokkan lurus. Setelah semua dikelompokkan hasil jumlah barang dan total barang secara keseluruhan pun tampil pada layar LCD sesuai dengan jenis barang apa yang terdeteksi.

Setelah semua proses selesai, langkah selanjutnya yaitu mematikan mesin konveyor dengan cara *logout* terlebih dahulu. Sebelumnya kita harus membuat program terlebih dahulu dan diupload. Program terdapat pada Gambar 4.48.

```

//-----LOGOUT
else if (proses == -101) //LOGOUT
{
    lcd.setCursor(0, 1);
    lcd.print(" Place Your Finger ");
    lcd.setCursor(0, 2);
    lcd.print(" to Logout ");
    //lcd.setCursor(0, 2);
    //lcd.print(myNIP);
    if (digitalRead(pcbol2) == LOW)
    {
        proses = 0;
        lcd.clear();
    }

    int myFID = getFingerprintIDez();

    if (myFID > 0) //ada finger terdaftar terdeteksi
    {
        //    lcd.setCursor(0, 3);
        //    lcd.print(myNIP);
        //    lcd.setCursor(5, 3);
        //    lcd.print(myFID);
        //    lcd.print(" ");
        if (myFID < start_ID_finger_ADMIN)
        {
            myFID = map(myFID, 1, 21, 1, 11); //di konversikan
        }
    }
}

```

Gambar 4.48 Program untuk *Logout*

Setelah *terupload*, maka untuk *logout* kita harus menekan tombol *push button* paling kanan yang digunakan untuk memilih menu yang ada pada LCD dengan tulisan “*Logout*” . Kemudian akan muncul perintah meletakkan sidik jari pada sensor *fingerprint* untuk melakukan *logout*, yang dapat dilihat pada Gambar 4.49



Gambar 4.49 Tampilan Perintah *Logout* pada LCD

Setelah itu meletakkan sidik jari yang sama pada sensor *fingerprint* untuk proses *logout*. Ketika sidik jari terdeteksi maka *logout* pasti berhasil dan pemberitahuan ketika *logout* berhasil akan ditampilkan pada LCD seperti pada Gambar 4.50



Gambar 4.50 Tampilan Ketika *Logout* Berhasil

Pada Gambar 4.48 menunjukkan bahwa hasil yang didapatkan sudah sesuai dengan program yang kami buat. Hasil deteksi sidik jari saat *logout* pun sudah sesuai karena untuk *login* juga menggunakan sidik jari “ADMIN”. Setelah *Logout* Berhasil maka mesin konveyor mati (tidak bisa berjalan).

Data saat *login* dan *logout* akan tersimpan secara otomatis pada *software* PLX DAQ. Untuk dapat menggunakan PLX DAQ ini harus memprogram terlebih dahulu supaya data dapat tersimpan pada PLX DAQ. Program dapat dilihat pada Gambar 4.51

```
//-----DAQ
Serial.print("DATA,DATE,TIME,LOGOUT"); Serial.print(",");
Serial.print(NIPDaQ); Serial.print(",");
Serial.print(nQrB); Serial.print(",");
Serial.print(nQrM); Serial.print(",");
Serial.print(nQrS); Serial.print(",");
Serial.print(nQrT); Serial.print(",");

Serial.print(nStB); Serial.print(",");
Serial.print(nStM); Serial.print(",");
Serial.print(nStS); Serial.print(",");
Serial.print(nStT);

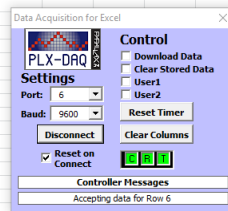
Serial.print("\n");

nQrB = 0; //301-399
nQrM = 0; //201-299
nQrS = 0; //101-199
nQrT = 0;
//Tiwlk
nStB = 0;
nStM = 0;
nStS = 0;
proses = -2;
}
```

Gambar 4.51 Program Menampilkan Data pada PLX DAQ

Setelah program terupload, Buka PLX DAQ lalu sesuaikan *Port* dan *Baudrate* yang digunakan pada Arduino. Setelah PLX DAQ *diconnectkan* maka tampil hasil data *login* dan *logout* yang dapat dilihat pada Gambar 4.52

A	B	C	D	E	F	G	H	I	J	K	L
Date	Time	Status	NIP	nQB	nQM	nQS	nQT	nSB	nSM	nSS	nST
6/27/2019	8:36:51 PM	LOGIN	ADMIN	0	0	0	0	0	0	0	0
6/27/2019	8:39:58 PM	LOGOUT	ADMIN	0	0	0	0	1	1	1	3
6/27/2019	8:40:01 PM	LOGIN	ADMIN	0	0	0	0	0	0	0	0
6/27/2019	8:41:26 PM	LOGOUT	ADMIN	0	0	0	0	1	1	1	2
6/27/2019	8:41:29 PM	LOGIN	1110001	0	0	0	0	0	0	0	0
6/27/2019	8:43:45 PM	LOGOUT	1110001	0	0	0	0	1	1	2	4



Gambar 4.52 Data Tersimpan pada PLX DAQ

Berdasarkan Gambar 4.52 data yang tersimpan pada PLX DAQ sudah sesuai dengan perancangan pendataan barang. Karena terlihat bahwa ketika karyawan sedang *login* untuk mengaktifkan mesin konveyor dan *logout* untuk mematikan mesin konveyor, maka

datanya otomatis tersimpan pada PLX DAQ. Serta jumlah barang yang telah dikelompokkan sesuai dengan jenis barang maka otomatis tersimpan juga. Jadi saat *login* data berupa tanggal, waktu *login*, NIP karyawan, dan jumlah barang. Untuk jumlah barang pada saat *login* totalnya masih 0 karena belum melakukan proses pengelompokkan barang. Pada saat *logout* jumlah barang sudah muncul karena telah melakukan proses pengelompokkan sehingga terlihat jumlah dari masing-masing bentuk barang dan total keseluruhan. Ketika *logout* data berupa tanggal, waktu, NIP serta jumlah barang akan otomatis tersimpan. Untuk jumlah barang pada Gambar 4.52 dilihat pada kolom nQB = jumlah barang besar berdasarkan QR Code, nQM = jumlah barang sedang berdasarkan QR Code, nQS = jumlah barang kecil berdasarkan QR Code, nQT = total keseluruhan barang berdasarkan QR Code, nSB = jumlah barang besar berdasarkan sensor ketinggian, nSM = jumlah barang sedang berdasarkan sensor ketinggian, nSS = jumlah barang kecil berdasarkan sensor ketinggian, nST = total keseluruhan barang berdasarkan sensor ketinggian. Pada Gambar 4.52 terlihat bahwa untuk jumlah barang berdasarkan Qr Code tetap 0 karena tidak melakukan pengelompokkan barang berdasarkan Qr Code. Tetapi hanya melakukan pengelompokkan barang berdasarkan ketinggian (dideteksi sensor dioda laser). Maka yang muncul jumlah barang hanya nSB, nSM, nSS, nST . Dengan ini maka pendataan barang lebih efektif karena tidak perlu untuk mendata barang satu persatu.

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Bab ini berisi tentang kesimpulan-kesimpulan yang didapatkan selama proses pembuatan Proyek Akhir ini beserta saran-saran untuk perbaikan dan pengembangannya.

5.1 Kesimpulan

Berdasarkan pengujian dan analisa Penggunaan Sensor *Fingerprint* Sebagai Aktivasi Mesin dan Pendataan Barang Secara Otomatis diperoleh beberapa kesimpulan diantaranya :

1. Sensor *Fingerprint* dapat mendeteksi posisi sidik jari lurus, miring kiri, maupun miring kanan.
2. Sidik jari dapat terdeteksi dengan baik apabila jari dalam kondisi bersih dibandingkan dengan kondisi tangan yang berminyak.
3. Persentase keberhasilan dari sensor *fingerprint* yaitu sebesar 92,5 %.
4. Persentase keberhasilan sensor dioda laser untuk mendeteksi barang yaitu sebesar 100% karena barang dikelompokkan sesuai jenis barang dengan tepat dan sesuai.
5. Jarak dari sensor dioda laser ke barang yang dideteksi tidak mempengaruhi saat proses pendeteksian barang oleh sensor dioda laser.
6. Sensor dioda laser dapat mendeteksi dengan baik dan tepat karena saat beberapa kali melakukan pengujian, sensor dioda laser selalu mendeteksi dan barang dapat dikelompokkan dengan tepat sesuai dengan jenis barang.
7. Berat barang kecil yang menaiki *lift* idealnya yaitu 14 gram, dikarenakan semakin berat barang tersebut maka akan mempengaruhi kecepatan konveyor saat naik keatas *lift*.
8. Data saat *login* dan *logout* dapat tersimpan secara otomatis pada *software* PLX DAQ

Proyek Akhir ini merupakan lanjutan dari Proyek Akhir Tahun 2018. Yang mempunyai perbedaan dengan Proyek Akhir sebelumnya yaitu untuk aktivasi mesinnya menggunakan sensor *fingerprint* dan

jumlah barang selain ditampilkan pada LCD juga tersimpan secara otomatis pada PLX DAQ.

5.2 Saran

Saran–saran yang dapat diberikan untuk implementasi dan pengembangan lebih lanjut dari sistem ini :

1. Untuk kedepannya sistem ini bisa dikembangkan dengan berbasis IOT (*Internet Of Things*) sehingga proses penyimpanan data bisa ditampilkan pada *website* sehingga dapat dipantau dari mana saja dan dapat mudah untuk diakses.
2. Untuk aktivasi mesin dapat dikembangkan dengan *face detection*.
3. Untuk pengelompokkan barang, saat *input* barang dapat dilakukan secara bersamaan. Sehingga proses pengelompokkan dapat berlangsung dengan cepat dan tidak menunggu barang satu persatu untuk dikelompokkan.
4. Untuk kedepannya *controller* bisa diganti yang lebih baik lagi.

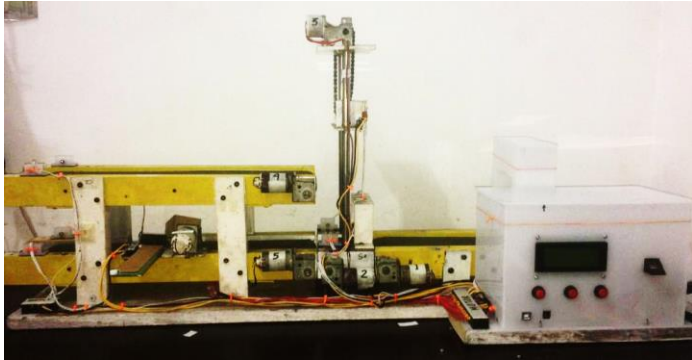
DAFTAR PUSTAKA

- [1] Andik Kurnia Adi Pratama dan Rahmat Bagus Prasajo, “Mesin Pengelompok Barang Berdasarkan Ketinggian Dan Konveyor Pengangkut Menggunakan PLC”, **Tugas Akhir**, Program Studi D3 Teknik Elektro FTI-ITS, Surabaya, 2015.
- [2] Muhammad Saiful Anwar dan Agam Abdillah, “Sistem Pengaman Pintu Rumah Menggunakan *Fingerprint* Scanner berbasis Mikrokontroler”, **Tugas Akhir**, Program Studi D3 Teknik Elektro FTI-ITS, Surabaya, 2016.
- [3] Kho Dickson, **Pengertian Dioda Laser dan Aplikasinya**, <https://teknikelektronika.com/pengertian-dioda-laser-aplikasi-simbol-laser-diode/> , 25 Mei 2019
- [4] Mochammad Fajar Wicaksono dan Hidayat, **Mudah Belajar Mikrokontroler Arduino**, Penerbit Informatika, Bandung, 2017.
- [5] Nuva Choironi Ersha, “Monitoring Program Pengelompokkan dan Penghitung Jumlah Barang Pada Konveyor Menggunakan Mikrokontroler”, **Tugas Akhir**, Surabaya, 2018.
- [6] Aristawati Dhiyan Widodo, “Monitoring Perhitungan dan Pengelompokkan Barang di Konveyor Berdasarkan Kode Batang (Scanner Barcode) Melalui LCD, **Tugas Akhir**, Program D3 Teknik Elektro FTI-ITS, Surabaya, 2018.
- [7] Aswindha Nasrullah, “Sistem Lock Door Panel Listrik Menggunakan *Fingerprint* Scanner Berbasis Android”, **Tugas Akhir**, Program Studi D3 Teknik Elektro FTI-ITS, Surabaya, 2017.
- [8] Kadir, Abdul., **Buku Pintar Pemrograman Arduino**, MediaKom, Yogyakarta, 2015.
- [9] Syarif Hidayatullah, “Rancang Bangun Alat Penghitung Jumlah Produksi Semen Berbasis Data Logger Menggunakan Sensor Infrared”, **Tugas Akhir**, Program D3 Teknik Elektro FTI-ITS, Surabaya, 2010.

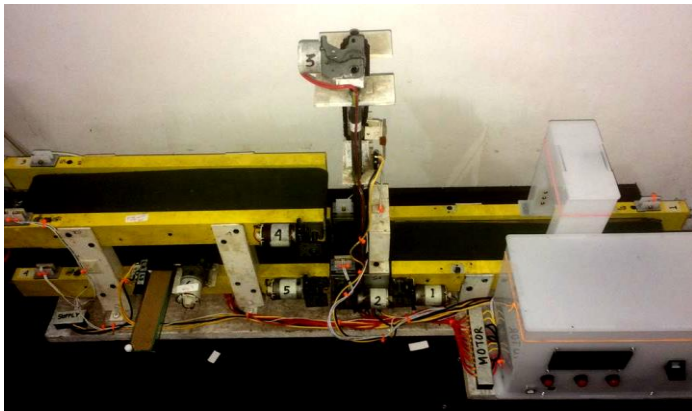
-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A

A.1. Foto Alat Secara Keseluruhan



Gambar Alat Secara Keseluruhan Tampak Depan



Gambar Alat Secara Keseluruhan Tampak Atas



Gambar Alat Secara Keseluruhan Ketika Dinyalakan

A.2. Foto Panel *Box* dan Model Barang



Gambar Panel *Box* Ketika *Box* Dinyalakan

LAMPIRAN B

B.1 Pemrograman Arduino untuk Validasi Finger

```
#include <Adafruit_Fingerprint.h>
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial2);
unsigned char status_hw_finger = 0; //0=ERROR , 1=OK
void setup_finger()
{
  finger.begin(57600);
  if (finger.verifyPassword()) //Check Finger
  {
    status_hw_finger = 1;
  }
  else
  {
    status_hw_finger = 0;
  }
  finger.getTemplateCount(); //load data finger yang sudah
terekam
  //Serial.print("Sensor          contains          ");
  Serial.print(finger.templateCount); Serial.println(" templates");
  //Serial.println("Waiting for valid finger...");
}
uint8_t getFingerprintID()
{
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
```

```

        break;
    case FINGERPRINT_NOFINGER:
        Serial.println("No finger detected");
        return p;
    case FINGERPRINT_PACKETRECEIVEERR:
        Serial.println("Communication error");
        return p;
    case FINGERPRINT_IMAGEFAIL:
        Serial.println("Imaging error");
        return p;
    default:
        Serial.println("Unknown error");
        return p;
}

// OK success!
p = finger.image2Tz();
switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Image converted");
        break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Image too messy");
        return p;
    case FINGERPRINT_PACKETRECEIVEERR:
        Serial.println("Communication error");
        return p;
    case FINGERPRINT_FEATUREFAIL:
        Serial.println("Could not find finger features");
        return p;
}

```

```

case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
default:
    Serial.println("Unknown error");
    return p;
}
// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK)
{
    Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}
// found a match!
//Serial.print("Found ID #"); Serial.print(finger.fingerID);
//Serial.print("      with      confidence      of      ");
Serial.println(finger.confidence);
return finger.fingerID;
}
// returns -1 if failed, otherwise returns ID #

```

```

int getFingerprintIDez() {
    uint8_t p = finger.getImage();
    if (p != FINGERPRINT_OK) return -1;
    p = finger.image2Tz();
    if (p != FINGERPRINT_OK) return -1;
    p = finger.fingerFastSearch();
    if (p != FINGERPRINT_OK) return -1;
    // found a match!
    //Serial.print("Found ID #"); Serial.print(finger.fingerID);
    //Serial.print("        with        confidence        of        ");
    Serial.println(finger.confidence);
    return finger.fingerID;
}

```

B2. Pemrograman untuk Menambahkan Sidik Jari Baru

```

else if (proses == -21)
{
    lcd.setCursor(0, 0);
    lcd.print("Select NIP");
    lcd.setCursor(0, 1);
    lcd.print("NIP="); lcd.print(nip[index_nip] );
    lcd.setCursor(0, 3);
    lcd.print("    Back    Next");
    if (digitalRead(ptombol2) == LOW)
    {
        proses = -2;
    }
    if (digitalRead(ptombol1) == LOW)

```

```

{
    index_nip++;
    if (index_nip > 10)
        index_nip = 1;
    Serial.println(index_nip);
    delay(300);
}
if (digitalRead(ptombol3) == LOW)
{
    proses = -211;
    lcd.clear();
}

}

else if (proses == -211) //ADD Finger
{
    index_nip_id = 1 + (index_nip - 1) * 4;
    lcd.setCursor(0, 0);
    lcd.print(" Place Your Finger ");
    lcd.setCursor(0, 1);
    lcd.print("  (Admin)  ");
    proses = -212;
}

else if (proses == -212) //ADD Finger
{
    int myFID = getFingerprintIDez();
    if (myFID > start_ID_finger_ADMIN) //Admin Finger Addresss
123-127
    {

```

```

    lcd.setCursor(0, 0);
    //lcd.print("01234567890123456789");
    lcd.print("    Finger 1    ");
    lcd.setCursor(0, 1);
    lcd.print("    Place Finger    ");
    proses = -213;
}
delay(50);
}
else if (proses == -213)
{
    id = index_nip_id;
    while (! getFingerprintEnroll() );
    delay(3000);
    lcd.setCursor(0, 0);
    //lcd.print("01234567890123456789");
    lcd.print("    Finger 2    ");
    lcd.setCursor(0, 1);
    lcd.print("    Place Finger    ");
    id++;
    while (! getFingerprintEnroll() );
    delay(3000);
    id = index_nip_id;
    lcd.setCursor(0, 0);
    //lcd.print("01234567890123456789");
    lcd.clear();
    lcd.print("    Finger    ");
    lcd.setCursor(0, 1);

```

```

lcd.print("  NIP="); lcd.print(nip[index_nip]);
lcd.setCursor(0, 3);
lcd.print("  Save Success  ");
delay(2000);
index_nip_id = 0;
index_nip = 0;
proses = -2;
}

```

B.3. Pemrograman untuk *Login*

```

else if (proses == -1)
{
  if (digitalRead(ptombol1) == LOW) //Check Sensor
  {
    proses = -11;
    lcd.clear();
  }
  if (status_hw_finger == 1)
  {
    if (digitalRead(ptombol3) == LOW) //Add User
    {
      proses = -21;
      delay(1000);
      lcd.clear();
    }
  }
  if (status_hw_finger == 0)
  {

```

```

if (digitalRead(ptombol2) == LOW) //back to login page
{
    proses = 0;
    lcd.clear();
}
}
String NIPdaQ = "";
int myFID = getFingerprintIDez();
//Serial.println(myFID);
if (myFID > 0) //Ada finger terdeteksi dan sudah pernah disimpan
{
    lcd.clear();
    //lcd.print("01234567890123456789");
    lcd.setCursor(0, 0);
    lcd.print("    NIP    ");
    if (myFID < start_ID_finger_ADMIN) //ID finger bukan ID
Admin, Admin 126-127
    {
        myNIP = map(myFID, 1, 21, 1, 11);
        lcd.setCursor(6, 1);
        lcd.print(nip[myNIP]);
        NIPdaQ = nip[myNIP];
    }
else
{
    lcd.setCursor(6, 1);
    lcd.print(" ADMIN ");
    NIPdaQ = "ADMIN";
    myNIP = 0;
}

```

```

}
lcd.setCursor(0, 2 );
lcd.print("  Login Success  ");
lcd.setCursor(0, 3 );
lcd.print("  Please wait  ");

```

B.4. Pemrograman untuk Menampilkan Data pada PLX DAQ

```

Serial.print("DATA,DATE,TIME,LOGIN"); Serial.print(",");
  Serial.print(NIPdaQ); Serial.print(",");
  Serial.print(nQrB); Serial.print(",");
  Serial.print(nQrM); Serial.print(",");
  Serial.print(nQrS); Serial.print(",");
  Serial.print(nQrT); Serial.print(",");

  Serial.print(nStB); Serial.print(",");
  Serial.print(nStM); Serial.print(",");
  Serial.print(nStS); Serial.print(",");
  Serial.print(nStT);
  Serial.print("\n");

  delay(2000);
  lcd.clear();
  proses = 0;
}
if (myFID == -3)
{
  lcd.clear();
  lcd.setCursor(0, 1 );

```

```

    lcd.print("  Login Failed  ");
    lcd.setCursor(0, 2 );
    lcd.print("  Please wait  ");
    delay(1000);
    proses=-2;
}
delay(50);
}

```

B.5. Pemrograman untuk Mengelompokkan Barang Berdasarkan Sensor Dioda Laser

```

//-----BIG
else if (benda == big && proses == 3)
{
    //motor(motor_1_k_depan, nyala);
    motor(motor_2_k_lift, nyala);
    dsensor_2 = baca_sensor(psensor_2);
    if (dsensor_2 == sensor_terhalang)
    {
        motor(motor_5_k_bawah, nyala);
        motor(motor_1_k_depan, mati);
        delay(1000);//untuk memastikan barang sudah berpindah ke
konveyor 3
        motor(motor_2_k_lift, mati);
        proses = 331;
    }
}
else if (proses == 331)
{
    dsensor_4 = baca_sensor(psensor_4);

```

```

if (dsensor_4 == sensor_terhalang)
{
    delay(1000);
    motor(motor_5_k_bawah, mati);
    proses = 0; inputString = "";
}
}
//-----MEDIUM
else if (benda == medium && proses == 3)
{
    motor(motor_1_k_depan, nyala);
    motor(motor_2_k_lift, nyala);
    dsensor_2 = baca_sensor(psensor_2);
    if (dsensor_2 == sensor_terhalang)
    {

        motor(motor_5_k_bawah, nyala);
        motor(motor_1_k_depan, mati);
        delay(1000);
        motor(motor_2_k_lift, mati);
        proses = 321;
    }
}
else if (proses == 321)//proses barang medium di konveyor 3
{
    delay(1700);
    motor(motor_5_k_bawah, mati);
    proses = 322;
}

```

```

else if (proses == 322)//proses barang medium didorong
{
    motor_push_dorong();
    if (dlimitdorong_belakang == 1)
        proses = 323;
}
else if (proses == 323)//proses pusher mundur
{
    motor_push_tarik();
    if (dlimitdorong_depan == 1)
        proses = 0; inputString = "";
}
//-----SMALL
else if (benda == small && proses == 3)//proses setelah sortir
{
    motor(motor_1_k_depan, nyala);
    motor(motor_2_k_lift, nyala);
    dsensor_2 = baca_sensor(psensor_2);
    if (dsensor_2 == sensor_terhalang)
    {
        motor(motor_2_k_lift, mati);
        motor(motor_1_k_depan, mati);
        delay(1000);
        proses = 311;

        timeup = millis();
        Serial.print("START :");
        Serial.print(timeup);
        Serial.print(" ms \n");
    }
}

```

```

    }
}
else if (proses == 311)//lift mulai naik
{
    lift_naik();
    if (dlimit/lift_atas == 0)
    {
        unsigned long time_done = millis();
        Serial.print("STOP :");
        Serial.print(time_done);
        Serial.print(" ms \n");

        Serial.print("DIF :");
        Serial.print(time_done - timeup);
        Serial.print(" ms \n");

        delay(200);//lift berhenti setelah mengenai limit atas dan berhenti
        dulu selama 200 ms
        motor(motor_2_k_lift, nyala);
        motor(motor_4_k_atas, nyala);
        delay(800);
        motor(motor_2_k_lift, mati);
        proses = 312;
    }
}
else if (proses == 312)
{
    lift_turun();

```

```

dsensor_3 = baca_sensor(psensor_3);
if (dsensor_3 == sensor_terhalang)
{
    flag_small = 1; //tanda bahwa barang kecil sudah mengenai sensor 3
}
if (dlimitlift_bawah == 0 && flag_small == 1)
{
    motor(motor_4_k_atas, mati);
    proses = 0; inputString = "";
    flag_small = 0;
    itunggu_small = 0;
}
if (flag_small == 1)
{
    itunggu_small++;
    if (itunggu_small > 440)
    {
        motor(motor_4_k_atas, mati);
    }
}
}

```

B.6. Pemrograman untuk *Logout*

```

else if (proses == -101) //LOGOUT
{
    lcd.setCursor(0, 1);
    lcd.print(" Place Your Finger ");
    lcd.setCursor(0, 2);

```

```

lcd.print("    to Logout    ");
//lcd.setCursor(0, 2);
//lcd.print(myNIP);
if (digitalRead(ptombol2) == LOW)
{
    proses = 0;
    lcd.clear();
}

int myFID = getFingerprintIDez();

if (myFID > 0) //ada finger terdaftar terdeteksi
{
    //    lcd.setCursor(0, 3);
    //    lcd.print(myNIP);
    //    lcd.setCursor(5, 3);
    //    lcd.print(myFID);
    //    lcd.print(" ");
    if (myFID < start_ID_finger_ADMIN)
    {
        myFID = map(myFID, 1, 21, 1, 11); //di konversikan
    }
    else
    {
        myFID = 0; // punya admin
    }
    //    lcd.setCursor(10, 3);
    //    lcd.print(myFID);
    //    delay(3000);

```

```

String NIPdaQ = "";
if (myNIP == myFID || myFID == 0) //Bisa dilogoutkan
admin
{

    myNIP = myFID;
    lcd.clear();
    //lcd.print("01234567890123456789");
    lcd.setCursor(0, 0);
    lcd.print("    NIP    ");
    if (myNIP == 0 ) //ADMIN
    {
        lcd.setCursor(6, 1);
        lcd.print(" ADMIN ");
        NIPdaQ = "ADMIN";
    }
    else
    {
        lcd.setCursor(6, 1);
        lcd.print(nip[myNIP]);
        NIPdaQ = nip[myNIP];
    }

    lcd.setCursor(0, 2 );
    //lcd.print("01234567890123456789");
    lcd.print(" Logout Success ");
    lcd.setCursor(0, 3 );

```

```
lcd.print(" Please wait ");  
delay(2000);  
lcd.clear();
```

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN C

C.1. Datasheet *Fingerprint* DY50 (FPM10A)

Fingerprint Sensor Module with Arduino (FPM10A)

Introducing the Fingerprint Sensor Module in the following figure, made fingerprint recognition more accessible and easy to add to your projects.

This means that it is super easy to make fingerprint collection, registration, comparison and search.



These modules come with FLASH memory to store the fingerprints and work with any microcontroller or system with TTL serial. These modules can be added to security systems, door locks, time attendance systems, and much more. Prices for this sensor greatly vary from \$10.

Specifications

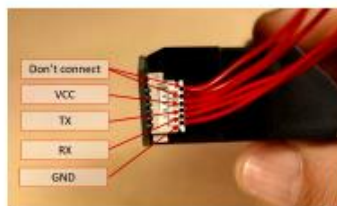
Voltage supply: DC 3.6 to 6.0V, Current supply: <120mA, Backlight color: green, Interface: UART

Bad rate: 9600, Safety level: five (from low to high: 1, 2, 3, 4, 5), False Accept Rate (FAR): <0.001% (security level 3)

False Reject Rate (FRR): <1.0% (security level 3), Able to store 127 different fingerprints.

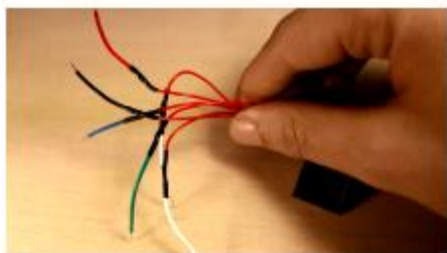
Sensor Pinout

The sensor has six pins that are labeled in the figure below.



The fingerprint sensor module used in this project came with really thin wires, so soldering breadboard-friendly wires was needed. We recommend using different colors according to the pin function. In our case:

DNC - white wire
VCC - red wire
TX - blue wire
RX - green wire
GND - black wire



The following table shows how to wire the sensor to the Arduino.

Fingerprint Sensor	Arduino
VCC	5V (it also works with 3.3V)
TX	RX (digital pin 2, software serial)
RX	TX (digital pin 3, software serial)
GND	GND

Installing the Adafruit Fingerprint Sensor Library

The easiest way to control the fingerprint sensor module with the Arduino is by using the Adafruit library for this sensor. Follow the next instructions to install the library:

1. Click here to download the [Adafruit Fingerprint Sensor Library](#)
2. Unzip the .zip folder and you should get Adafruit-Fingerprint-Sensor-Library-master folder
3. Rename your folder from: ~~Adafruit-Fingerprint-Sensor-Library-master~~ folder to **Adafruit_Fingerprint_Sensor_Library** folder
4. Move the folder to your Arduino IDE installation libraries folder
5. Finally, re-open your Arduino IDE

Enroll a New Fingerprint

Having the fingerprint sensor module wired to the Arduino, follow the next steps to enroll a new fingerprint. Make sure you've installed the Adafruit Fingerprint Sensor library previously.

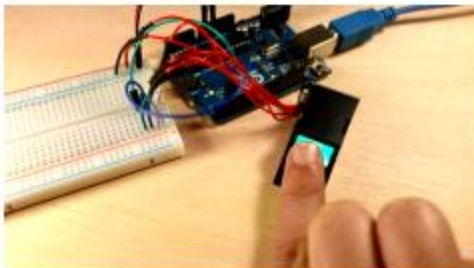
1. In the Arduino IDE, go to **File > Examples > Adafruit Fingerprint Sensor Library > [Finger1](#)**

2. Upload the code, and open the serial monitor at a baud rate of 9600.

3. You should enter an ID for the fingerprint. As this is your first fingerprint, type 1 at the top left corner, and then, click the **Send** button.



4. Place your finger on the scanner and follow the instructions on the serial monitor.



You'll be asked to place the same finger twice on the scanner.

If you get the **"Prints matched!"** message, as shown below, your fingerprint was successfully stored.

If not, repeat the process, until you succeed.



C.2. Datasheet AMS1117

**AMS1117**

0.8A Adjustable/Fixed Low Dropout Linear Regulator

General Description

The AMS1117-ADJ and AMS1117-1.2,-1.5,-1.8,-1.9,-2.5,-3.3 and-5.0 are low dropout three-terminal regulators with 0.8A output current capability. These devices have been optimized for low voltage where transient response and minimum input voltage are critical.

On-chip thermal limiting provides protection against any combination of overload and ambient temperatures that would create excessive junction temperatures.

The AMS1117 series regulators are available in the industry-standard SOT-89 packages.

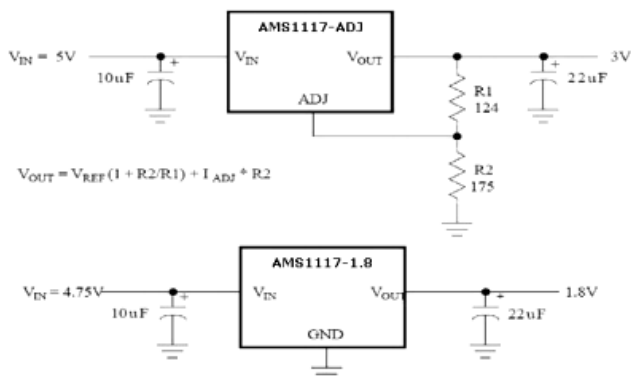
Key Features

- Low dropout voltage
- Load regulation: 0.2% typical
- Optimized for Low Voltage
- On-chip thermal limiting
- Standard SOT-89 packages
- Three-terminal adjustable or fixed low dropout 1.2V, 1.5V, 1.8V, 1.9V, 2.5V, 3.3V, 5V. Regulators

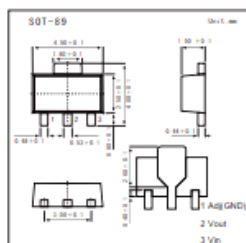
Applications

- High efficiency linear regulators
- Post regulators for switching supplies
- Battery chargers
- 12V to 5V linear regulators
- Motherboard clock supplies

Typical Application



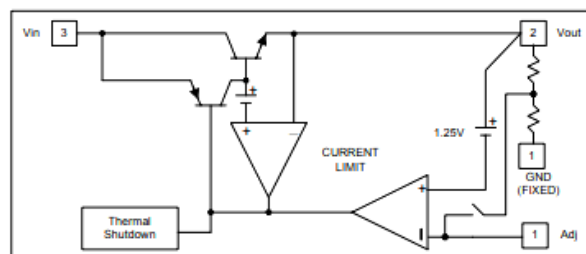
Typical Applications of AMS1117



■ Absolute Maximum Ratings $T_a = 25^\circ\text{C}$

Parameter	Symbol	Rating	Unit
Maximum Input Voltage	V_{in}	18	V
Power Dissipation	P_D	Internally Limited	
Operating Junction Temperature Range	T_J	150	$^\circ\text{C}$
Storage Temperature	T_{st}	-65 to +150	$^\circ\text{C}$

■ Block Diagram



■ Electrical Characteristics $T_a = 25^\circ\text{C}$

Parameter		Test conditions	Min	Typ	Max	Unit
Reference Voltage	V_{ref}	AMS1117-ADJ $10\text{mA} \leq I_{out} \leq 800\text{mA}$, $1.5\text{V} \leq V_{in}-V_{out} \leq 12\text{V}$	1.225	1.250	1.275	V
Output Voltage	V_{out}	AMS1117-1.2 $0 \leq I_{out} \leq 800\text{mA}$, $2.6\text{V} \leq V_{in}-V_{out} \leq 12\text{V}$	1.175	1.205	1.225	V
		AMS1117-1.5 $0 \leq I_{out} \leq 800\text{mA}$, $2.9\text{V} \leq V_{in}-V_{out} \leq 12\text{V}$	1.470	1.500	1.530	V
		AMS1117-1.8 $0 \leq I_{out} \leq 800\text{mA}$, $3.2\text{V} \leq V_{in}-V_{out} \leq 12\text{V}$	1.764	1.800	1.836	V
		AMS1117-1.9 $0 \leq I_{out} \leq 800\text{mA}$, $3.3\text{V} \leq V_{in}-V_{out} \leq 12\text{V}$	1.852	1.900	1.938	V
		AMS1117-2.5 $0 \leq I_{out} \leq 800\text{mA}$, $3.9\text{V} \leq V_{in}-V_{out} \leq 12\text{V}$	2.430	2.500	2.550	V
		AMS1117-3.3 $0 \leq I_{out} \leq 800\text{mA}$, $4.75\text{V} \leq V_{in}-V_{out} \leq 12\text{V}$	3.234	3.300	3.366	V
		AMS1117-5.0 $0 \leq I_{out} \leq 800\text{mA}$, $6.5\text{V} \leq V_{in}-V_{out} \leq 12\text{V}$	4.900	5.000	5.100	V
Line Regulation	ΔV_{out}	AMS1117-ADJ $I_{out}=10\text{mA}$, $1.5\text{V} \leq V_{in}-V_{out} \leq 13.775\text{V}$	0.005	0.2		%
		AMS1117-1.2 $I_{out}=10\text{mA}$, $2.6\text{V} \leq V_{in}-V_{out} \leq 12\text{V}$	9	12		mV
		AMS1117-1.5 $I_{out}=10\text{mA}$, $2.9\text{V} \leq V_{in}-V_{out} \leq 12\text{V}$	9	12		mV
		AMS1117-1.8 $I_{out}=10\text{mA}$, $3.2\text{V} \leq V_{in}-V_{out} \leq 12\text{V}$	9	12		mV
		AMS1117-1.9 $I_{out}=10\text{mA}$, $3.3\text{V} \leq V_{in}-V_{out} \leq 12\text{V}$	9	12		mV
		AMS1117-2.5 $I_{out}=10\text{mA}$, $3.9\text{V} \leq V_{in}-V_{out} \leq 12\text{V}$	9	12		mV
		AMS1117-3.3 $I_{out}=10\text{mA}$, $4.75\text{V} \leq V_{in}-V_{out} \leq 12\text{V}$	9	12		mV
Load Regulation	ΔV_{out}	AMS1117-5.0 $I_{out}=10\text{mA}$, $6.5\text{V} \leq V_{in}-V_{out} \leq 12\text{V}$	9	12		mV
		AMS1117-ADJ $V_{in}-V_{out}=5\text{V}$, $10\text{mA} \leq I_{out} \leq 800\text{mA}$	0.2	0.4		%
		AMS1117-1.2 $V_{in}=2.6\text{V}$, $0 \leq I_{out} \leq 800\text{mA}$	3	10		mV
		AMS1117-1.5 $V_{in}=2.9\text{V}$, $0 \leq I_{out} \leq 800\text{mA}$	3	10		mV
		AMS1117-1.8 $V_{in}=3.2\text{V}$, $0 \leq I_{out} \leq 800\text{mA}$	3	10		mV
		AMS1117-1.9 $V_{in}=3.3\text{V}$, $0 \leq I_{out} \leq 800\text{mA}$	3	10		mV
		AMS1117-2.5 $V_{in}=3.9\text{V}$, $0 \leq I_{out} \leq 800\text{mA}$	3	10		mV
Dropout Voltage	$V_{in}-V_{out}$	AMS1117-XXX $\Delta V_{out}, \Delta V_{ref}=1\%$, $I_{out}=500\text{mA}$	1.16	1.25		V
		AMS1117-XXX $\Delta V_{out}, \Delta V_{ref}=1\%$, $I_{out}=800\text{mA}$	1.26	1.3		V
		AMS1117-XXX $V_{in}-V_{out} = 5\text{V}$, $T_j=25^\circ\text{C}$	1.25	1.4	1.6	A
Current Limit	I_{out}	AMS1117-XXX	1.25	1.4	1.6	A
Minimum Load Current	I_{out}	AMS1117-XXX	5	10		mA
Quiescent current	I_q	AMS1117-ADJ $V_{in}-V_{out}=1.25\text{V}$	4	8		mA
		AMS1117-1.2 $V_{in}-V_{out}=1.25\text{V}$	4	8		mA
		AMS1117-1.5 $V_{in}-V_{out}=1.25\text{V}$	4	8		mA
		AMS1117-1.8 $V_{in}-V_{out}=1.25\text{V}$	4	8		mA
		AMS1117-1.9 $V_{in}-V_{out}=1.25\text{V}$	4	8		mA
		AMS1117-2.5 $V_{in}-V_{out}=1.25\text{V}$	4	8		mA
		AMS1117-3.3 $V_{in}-V_{out}=1.25\text{V}$	4	8		mA
Adjust Pin Current (Adjustable Version)	I_{adj}		55	120		μA
Adjust Pin Current Change	I_{change}		0.2			μA

■ Typical Characteristics

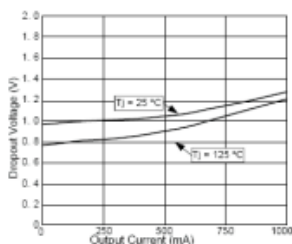


Fig.1 Dropout Voltage vs Output Current

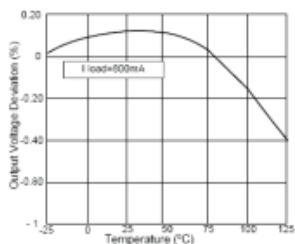


Fig.2 Load Regulation vs Temperature

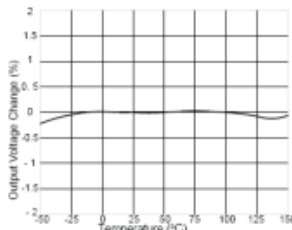


Fig.3 Percent Change in Output Voltage vs Temperature

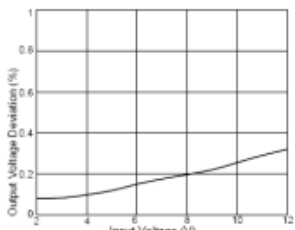


Fig.4 Line Regulation

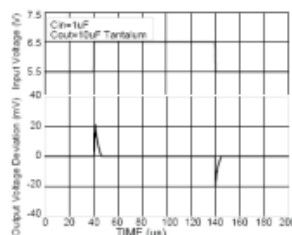


Fig.5 Line Transient Response

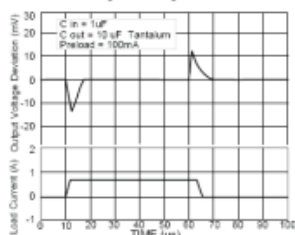


Fig.6 Load Transient Response



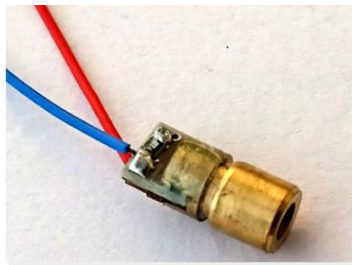
■ Ordering Information

Device	Packaging	Shipping
AMS1117-XX	SOT-89	1000/Tape&Reel

■ Marking

Marking	1117-XX
---------	---------

C.2. Datasheet Dioda Laser



Features

- RoHS (Restriction of Hazardous Substances) Compliant
- Quality level is high
- Cost is economical
- Wavelength from 635 nm to 660 nm
- Rise and fall time is 0.5ns
- Package available:- TO-18(dia. 5.6mm), TO-5 (dia. 9 mm)

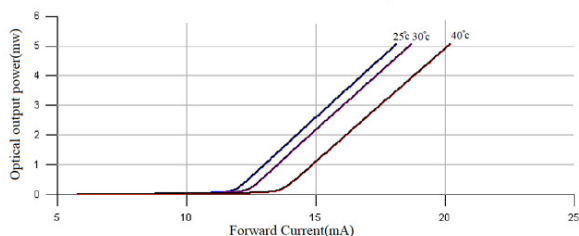
Technical Specification

- Operating Temperature:- -10°C to $+40^{\circ}\text{C}$
- Storage Temperature:- -15°C to $+85^{\circ}\text{C}$
- Output power (P_o):- 5 mW
- Normal and maximum operating voltage is 2.2 and 2.7 respectively
- Threshold current in minimum, normal and maximum condition are 15, 20 and 30mA
- Operating current is 65 to 80mA
- Beam Angle deviation:-
 - For both parallel and perpendicular condition its between -3 to 3 degree
- Beam divergence:-
 - For parallel condition it's between 8 to 12 degree
 - For perpendicular condition it's between 23 to 32 degree

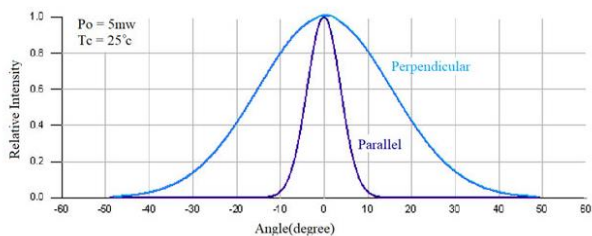
Characteristics Curves

1. Temperature Effect on Operation of Laser Diode

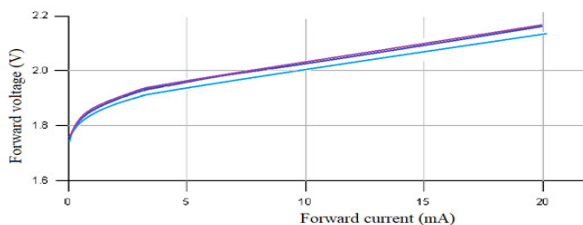
This graph is between Optical output power v/s forward current. It's clear from the graph that laser output will only be visible if obtained above the threshold value of the laser diode. Before the threshold value the output of the laser diode is zero. After the threshold value the output of laser diode increase with slightly increase in forward voltage. The effect of temperature in the operation of Laser Diode is shown in graph below:



2. Laser Beam Divergence in parallel and perpendicular plane

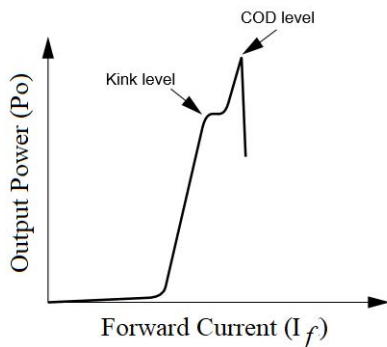


3. Forward voltage v/s Forward current



4. Output Power v/s Forward Current

If the direction of moving current is forward and the output is continuously increasing, after the kink level the laser face a sudden breakdown which is the COD (Catastrophic Optical Damage) level. At this level due to high optical density the crystal at the face of diode melts. At the time of manufacturing of Red lasers a special care is taken to avoid surge like static electricity and increase in current, because in red laser the oscillation is occur with the low power of 2 to 3 mW even after the breakdown. As the element is damaged the laser gets damaged or not able to work.



-----Halaman ini sengaja dikosongkan-----

LAMPIRAN D

DAFTAR RIWAYAT HIDUP



Nama : Pratiwi Maisita
Sari
TTL : Tulungagung, 26
Agustus 1998
Jenis Kelamin : Perempuan
Agama : Islam
Alamat Rumah : Jl. Gayungan
IV/23 Surabaya
Telpn/HP : 0863466293
E-mail : pratiwimaisita98
@gmaill.com

RIWAYAT PENDIDIKAN

- 2001 – 2003 : TK Nurul Huda
- 2004 – 2010 : SDN Gayungan 1/422 Surabaya
- 2010 – 2013 : SMPN 36 Surabaya
- 2013 – 2016 : SMAN 10 Surabaya
- 2016 – 2019 : Departemen Teknik Elektro
Otomasi Program Studi Teknologi
Otomasi - Fakultas Vokasi (FV)
Institut Teknologi Sepuluh
Nopember (ITS)

PENGALAMAN KERJA

- Magang di PT.PAL Indonesia (Persero) Agustus– Desember
2018